Communications of the Association for Information Systems

Volume 38 Article 29

5-2016

Resources-Events-Agents Design Theory: A Revolutionary Approach to Enterprise System Design

Cheryl Dunn
Grand Valley State University, justcheryl.dunn@gmail.com

Gregory J. Gerard
Florida State University

Severin V. Grabski
Michigan State University

Communications of the Association for Information Systems

Research Paper ISSN: 1529-3181

Resources-Events-Agents Design Theory: A Revolutionary Approach to Enterprise System Design

Cheryl Dunn
Grand Valley State University
School of Accounting
justcheryl.dunn@gmail.com

Gregory J. Gerard
Florida State University
Accounting

Severin V. Grabski
Michigan State University
Accounting & Information Systems

Abstract:

Enterprise systems typically include constructs such as ledgers and journals with debit and credit entries as central pillars of the systems' architecture due in part to accountants and auditors who demand those constructs. At best, structuring systems with such constructs as base objects results in the storing the same data at multiple levels of aggregation, which creates inefficiencies in the database. At worst, basing systems on such constructs destroys details that are unnecessary for accounting but that may facilitate decision making by other enterprise functional areas. McCarthy (1982) proposed the resources-events-agents (REA) framework as an alternative structure for a shared data environment more than thirty years ago, and scholars have further developed it such that it is now a robust design theory. Despite this legacy, the broad IS community has not widely researched REA. In this paper, we discuss REA's genesis and primary constructs, provide a history of REA research, discuss REA's impact on practice, and speculate as to what the future may hold for REA-based enterprise systems. We invite IS researchers to consider integrating REA constructs with other theories and various emerging technologies to help advance the future of information systems and business research.

Keywords: Resources-Events-Agents (REA), Design Theory, Enterprise Systems, Modeling, Database.

This manuscript underwent peer review. It was received 10/15/2014 and was with the authors for 4 months for 4 revisions. Ping Zhang served as Associate Editor.

1 Introduction

Resources-events-agents (REA) is a design theory for enterprise systems that is based on semantics, is incorporated into ISO standards for business exchange patterns, and serves as the basis for at least one cloud-based enterprise system. REA originated as a generalized framework to accommodate management information needs that the traditional accounting model did not adequately address (McCarthy, 1982). As REA research progressed from a generalized framework to a design theory, the accounting literature primarily published the related work, which rendered it relatively unknown by most mainstream IS researchers. In this paper, we document REA's origins and examine the REA literature through both a design science lens and a behavioral science lens. We also identify REA's contributions to research and practice and suggest future research avenues.

McCarthy (1982) originally introduced REA to overcome the limitations of traditional accounting systems because they did not provide much of the information necessary to run a business. Realizing the need for integrated enterprise systems, McCarthy used data modeling techniques and conceptual foundations of accounting theorists to develop the REA framework. This semantic framework addressed the traditional needs because one could derive the resultant financial statements and other reports from the primitives. REA preserved the duality of economic events (i.e., the causal relationship between the "gives" and the "takes"), identified the agents involved with these events (which is critical from a control perspective), and provided granular data about these events such that managers could obtain the data needed to enable effective decision making. The REA framework allowed one to include both planning data (e.g., purchase orders, employee training, and benefit programs) and historic data (e.g., sales). The REA framework provided some of the earliest theoretical evidence of how one could construct enterprise systems that integrate all of the planning, control, and communication functions across an organization.

Geerts and McCarthy (2000b, 2002) subsequently codified extensions to the original framework by incorporating type images, enterprise value chains, and workflow/task specification. Based on Gregor and Jones (2007) and consistent with Geerts, Graham, Mauldin, McCarthy, and Richardson (2013), we view REA as a design theory. Design theories such as REA focus on "how to do something" (Gregor & Jones, 2007, p. 313). REA provides a theoretically based design of inter-organizational exchange transactions and a foundation for inter-organizational enterprise systems that facilitate business-to-business e-commerce and the digitization of the enterprise. REA serves as the basis for the International Organization for Standardization/International Electrotechnical Commission standard 15944-4 on economic exchanges (ISO/IEC, 2007) and is the core data model architecture of two relatively new entrants into the enterprise system market (Workday and REA Technology, which we discuss in more detail later). Such use in practice is certainly evidence of a design theory that provides guidance of "how to do something." Research has reported that implementing REA results in significant savings in the total cost of ownership of the system and also results in improved user experience (Curry, 2009). Even legacy ERP systems' non-accounting modules, which have not been able to fully embrace REA because of the need to maintain compatibility for existing system users, are largely consistent with the REA theory (O'Leary, 2004; Fallon & Polovina, 2013).

This paper proceeds as follows. In Section 2, we summarize and provide examples of the REA design theory for those unfamiliar with its core concepts. In Section 3, we trace REA's intellectual heritage from its origin in the accounting literature to its position as a design theory in the information systems literature. Because many researchers focus their efforts in either the design science realm or the behavioral science paradigm, we summarize additional design science and behavioral studies separately in Sections 4 and 5, respectively. In Section 6, we identify REA's contributions to research and practice. In Section 7, we speculate about what the future may hold for REA. We suggest future research avenues and encourage interested readers to consider how integrating REA into their own research interests may be able to propel both forward.

2 Overview of REA Design Theory

McCarthy (1982) envisioned as a design objective a shared data environment from which all users in an enterprise could obtain the information they needed, at whatever level of aggregation they needed, whenever they needed it. He recognized that an accounting system structured as ledgers and journals with debit and credit entries rendered transaction-level data useless for non-accounting users because the data was aggregated too highly before being stored, possessed only accounting-relevant attributes, and was often commingled with accounting estimates or accruals. His proposal for a generalized framework included a shared data environment that would produce not just the information needed for accounting reports but also the information needed by marketing, production, logistics, and other enterprise decision makers. He named this generalized framework the resources-events-agents (REA) model. The original formulation of the model is based on transaction analyses, object pattern recognition, accounting theory, and economic theory (McCarthy, 1978). The REA model is semantic, technology independent, and reliant on robust theoretical principles, which makes it in essence timeless (Geerts, 2008). Since 1982, the scope of the REA model has extended to include commitments, policy specifications, strategic planning information needs, value chains, and supply chains. Individuals have applied REA in various industries and illustrated REA with semantic Web technologies, and REA has emerged as a robust approach for teaching accounting information systems courses (McCarthy, 2003; Geerts, 2008; Dunn, 2013). REA serves as the foundation for at least two software applications used in practice (Workday (http://www.workday.com) and REA Technology (http://www.reatechnology.com), which allows such software to fully trace planned, expected, and completed business transactions (Geerts, 2008).

2.1 REA Constructs

The general REA model for any business cycle comprises several components (presented in list form for brevity). Here, we overview the model at a high level; interested readers are encouraged to study McCarthy (1982), Geerts and McCarthy (2002, 2006), Hruby (2006), Dunn (2012), and Gailly and Geerts (2013) for more detail:

- *Economic events* represent alternative sides of an economic exchange or a conversion. Economic increment events increase at least one economic resource, and economic decrement events decrease at least one economic resource.
- *Economic resources* are the items received and given up in economic exchanges.
- *Internal agents* are members of the company's personnel responsible for the economic events (usually at least one agent type for each event).
- *External agents* are the people or companies with whom the company engages "at arms' length" in an exchange (often, but not always, the same external agent is connected to both causally related economic decrement and increment events).
- *Duality* is a relationship between economic increment and decrement events. An enterprise would not willingly engage in an economic decrement event unless it expects a related economic increment event to occur if not simultaneously, then at an agreed-on alternative time.
- *Stockflow* relationships represent the increases and decreases in quantity and/or value of resources as a result of economic increment and decrement events by connecting the economic events to the associated resources.
- *Participation* relationships connect events to the agents (internal and external) who participate in the events.
- *Assignment* relationships designate agents of one category to work with agents of another category (e.g., salesperson assigned to customer). This relationship is direct and independent of any events in which the agents may mutually participate. The salesperson is assigned to the customer even if a sale is never made to the customer.
- *Custody* relationships connect resources to the agents who are accountable for them.

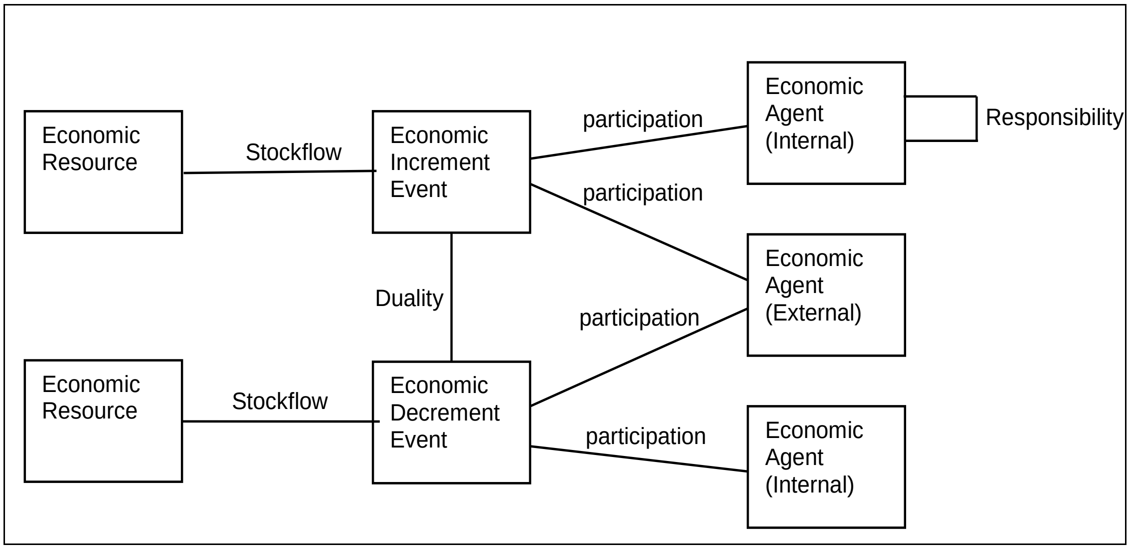

Figure 1. The REA Accounting Model (Adapted from McCarthy, 1982)[1]

The original REA model (see Figure 1) represented only the enterprise operational level, also called the accountability infrastructure, which is sufficient for historical accounting purposes. REA has evolved into a design theory, and it now incorporates the operational level (what has happened), commitment level (what is planned or scheduled), and the policy level (what could, should, or must happen). Extant REA design theory includes the following additional features and components (Geerts & McCarthy, 2002; 2006; Hruby, 2006; Dunn, 2012; Gailly & Geerts, 2013):

- *Value chain models* integrate transaction cycle models to a higher level of abstraction and connect the transaction cycles via the resources that flow between them.
- *Workflow or task level models* expand transaction cycle models to a lower level of abstraction and specify the steps or activities needed to accomplish the transaction cycle events.
- *Value system or supply chain models* denote the types of resource exchanges expected to occur between an enterprise and its external business partners and provide a high level overview of the enterprise business model.
- *Model components* are separated into continuants (enduring objects with stable attributes that allow them to be recognized on different occasions throughout a period of time) and occurrents (processes or events that are in a state of flux).
- *Duality relationships* are differentiated as transfer duality (in which a good is exchanged for another good or for cash) and transformation duality (in which a good is transformed into another good).
- *Type images* represent category level abstractions of similar components. For example, whereas an economic resource is a specifically identified physical object (an automobile with VIN 2CNALDEC1B6270317: only one exists), an economic resource type is a set of physical objects that have like characteristics (2011 Chevrolet Equinox: thousands of specifically identified instances of this type exist).
- *Commitments* represent agreements to engage in future economic events. Commitments may be further specified as increment commitments, decrement commitments, or mutual commitments (one commitment agrees to both a future increment and a future decrement).
- *Instigation events* represent whatever initiates the activities in a transaction cycle. Typically, these events identify some type of need and are expected to lead to commitments.

[1] As we discuss in this paper, REA design theory is independent of any particular modeling formalism. In this paper, we use simple UML class diagrams, whereas early work in REA has typically used ER diagrams.

- *Economic reversal* events (which may be specified as economic increment reversals or economic decrement reversals) are events that reverse economic events. Sale returns and purchase returns are examples of reversal events.
- *Fulfillment relationships* connect commitment events to the resulting economic events and connect instigation events to the resulting commitments.
- *Involvement relationships* connect economic events to the resource types that those events increase or decrease. These are less tangible than stockflow relationships given that resource types represent kinds of items rather than individually traceable items.
- *Reservation relationships* connect commitment events to the resources that are the proposed subjects of the future exchange.
- *Proposition relationships* connect instigation events to the resources that would become the proposed subjects of the future exchange
- *Typification relationships* connect resources, events, and agents to the categories to which they belong (i.e., resource-resource type, agent-agent type, and event-event type relationships).
- *Grouping relationships* connect resources, events, and agents to groups to which they are assigned. For example, an enterprise may organize its vehicles into fleets and create a grouping relationship from the vehicle resource to the fleet group.
- *Policy relationships* connect type images to communicate company policies (e.g., agent type-agent type, agent type-event type, agent type-resource type, etc.). For example, an enterprise may indicate a policy that says its wholesale sales may be made only to corporate customers whereas its retail sales may be made to any type of customer.
- *Specification relationships* connect commitments to the types of events, resources, and agents those commitments specify.
- *Return relationships* connect economic reversal events to the economic resources that those events returned.
- *Reversal relationships* connect economic reversal events to the original economic events that they reversed. Although such relationships connect two events (one of which increases a resource and the other of which decreases a resource), this relationship differs from duality in that both events involve the same resource rather than different resources. For example, sale and cash receipt are related via duality—the inventory resource is decreased and the cash resource is increased. Sale and sale return are related via reversal—the inventory resource is decreased and then increased. A sale return reverses a sale.
- *Reciprocal relationships* connect an increment commitment to a related decrement commitment. Sometimes a reciprocal relationships may be reified (converted to an entity), at which point it is defined as an *agreement*. If the agreement is for a future exchange (i.e., a transfer duality), it is called a *contract*. If the agreement is for a future conversion (i.e., a transformation duality), it is called a *schedule.*

Figure 2 illustrates the REA design theory for a single exchange or transaction cycle; one would create similar view models for each exchange/transaction cycle and then integrate them into an enterprise-wide model. An enterprise-wide model is clearly too large to show on a single page. In Figure 2, consistent with McCarthy (2010), we add colors to aid readers in distinguishing the various kinds of classes: policy infrastructure classes are yellow, scheduled or planned events are purple, and operational classes are green.

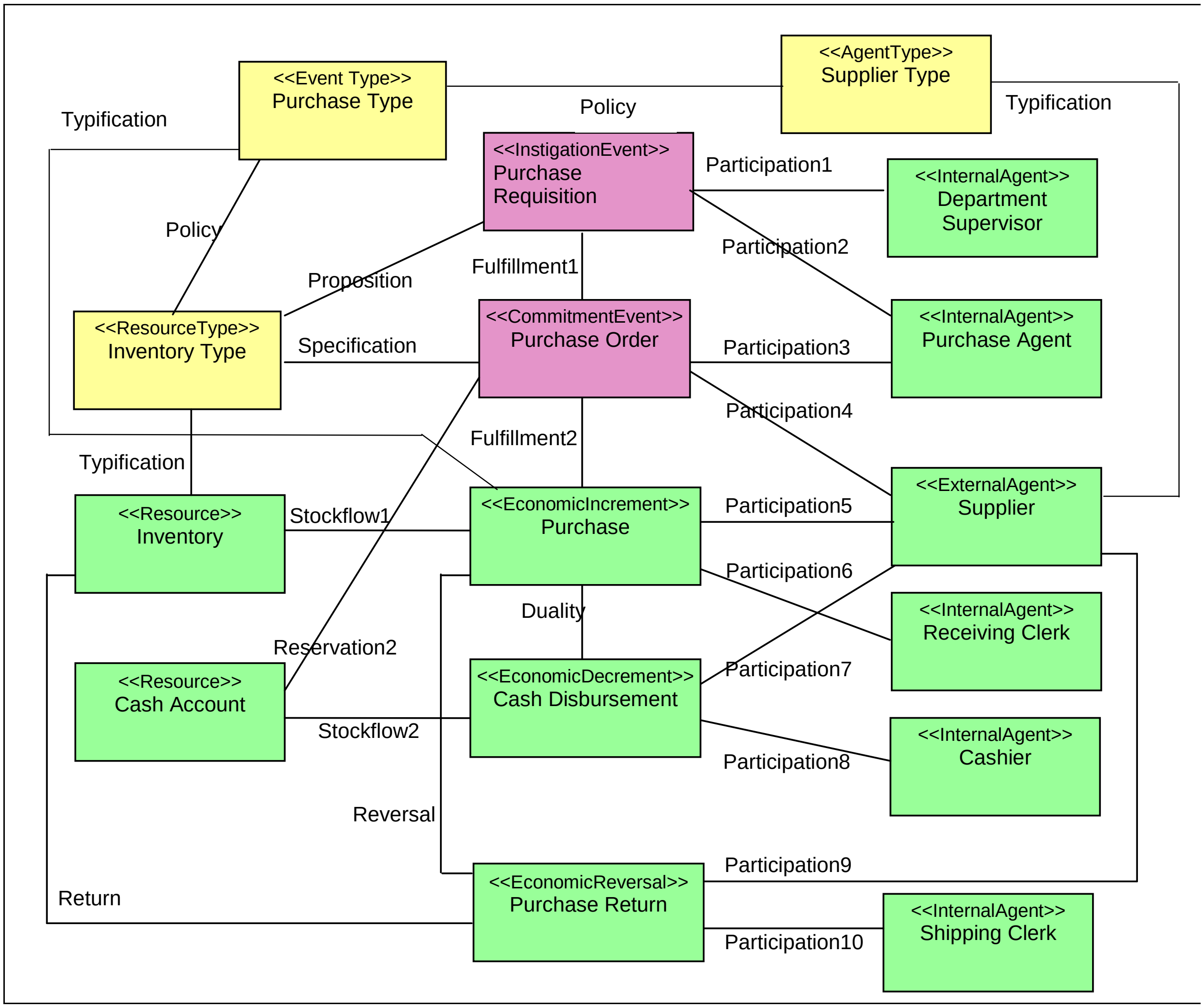

Figure 2. Expanded REA Design Theory Acquisition Cycle (Adapted from Dunn, 2012; McCarthy, 2010)

Besides the extensions made at the business process/transaction cycle level, REA has also been expanded to include value system-, value chain-, and workflow/task-level models. Value system REA models depict an enterprise in the context of its external business partners; these partners are the immediate links in the enterprise supply chain. Value chain REA models reflect an enterprise's script for doing business. The high-level business processes, also called transaction cycles (e.g., revenue, acquisition, conversion, financing, and human resources) that comprise an enterprise's value chain, are represented together with the resource flows that connect those processes. One could create additional diagrams to represent the workflow needed to accomplish each event in a business process, which may take the form of system flowcharts, unified modeling language (UML) activity or use-case diagrams, decision trees, or other suitable representations. Figure 3 illustrates the four levels of the REA design theory. The business process level and task level models expand on one acquisition exchange from the overall value chain; every exchange in the value chain would be modeled in a similar way at the business process and task levels. Furthermore, the business process level diagram, in practice, would be a rich and expressive UML class diagram; it would contain attributes, multiplicities, and so on. Once one had fully specified and integrated the business process level, one could transform it into logical and physical data models.

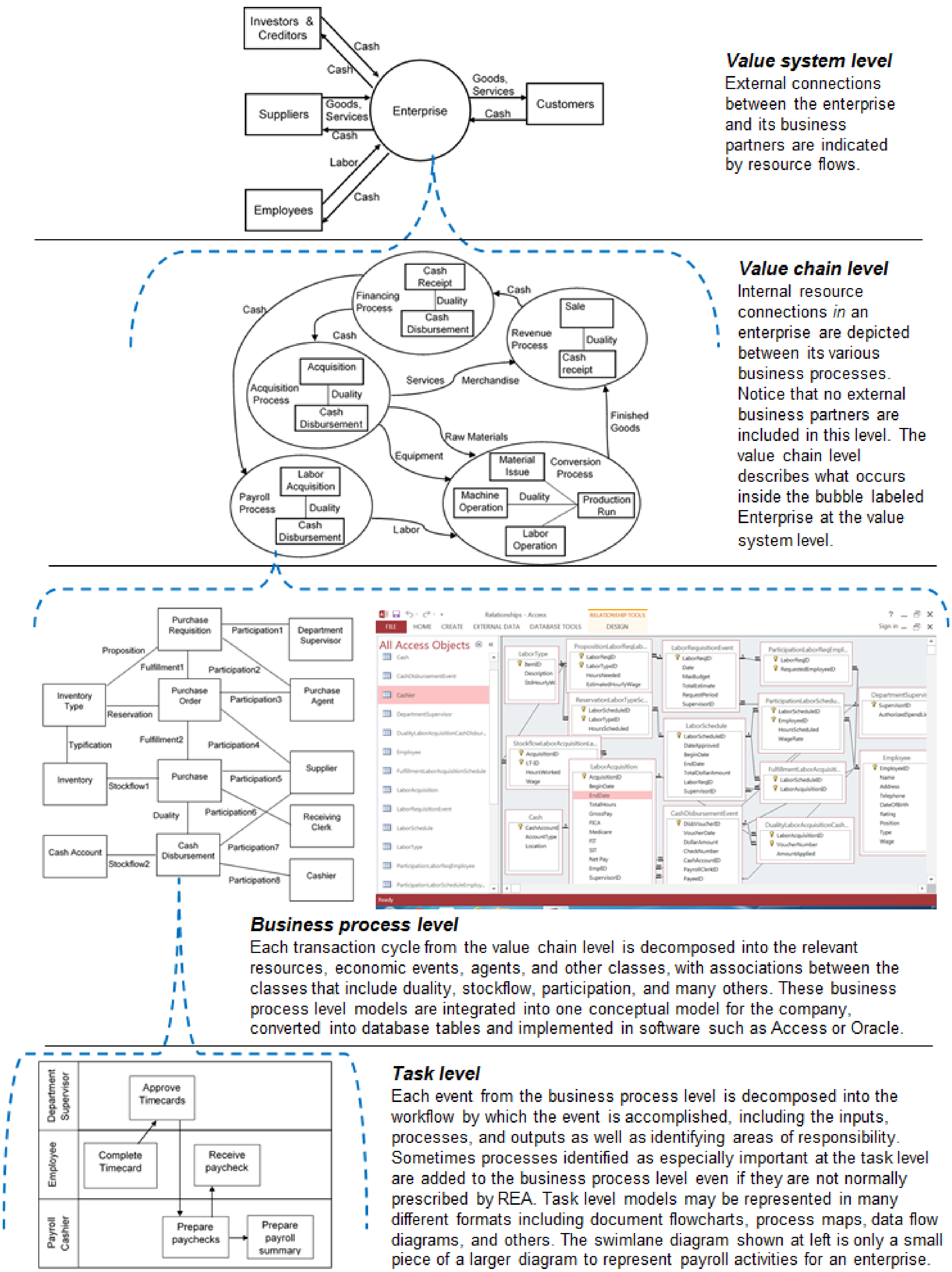

Figure 3. Four Levels of the REA Ontology (Adapted from Dunn, 2012)

3 Intellectual Heritage of REA Research

Based on the research of numerous scholars, REA evolved from a generalized accounting framework to an enterprise design theory over the years. Researchers with a strong accounting domain background conducted the initial REA research with the design science methodology and primarily in North America. Subsequently, researchers again with an accounting background conducted evaluative research again primarily in North America. Interest in REA grew and, currently, scholars in Europe conduct much of the design science and evaluative research on REA.

REA first appeared in McCarthy's (1978) doctoral dissertation, which he expanded on in subsequent publications (McCarthy, 1979, 1980, 1982). Around that time, some influential developments occurred in computing. First, E. F. Codd developed the theoretical basis for relational databases (see e.g., Codd, 1970). Second, Peter Chen published his work on the entity-relationship model (Chen, 1976). These advances supported the notion of data independence and led to the use of semantics in systems development. The explicit use of semantics is an essential feature of REA.

At the time that McCarthy (1979) appeared, discussion in the accounting literature focused on building accounting systems using hierarchical (Haseman & Whinston, 1976), network (Haseman & Whinston, 1977), and relational (Everest & Weber, 1977) database models. Instead of focusing on the logical- or physical-level models of accounting systems, McCarthy (1978) focused on the conceptual-level model and on representing reality with entities and relationships between entities. This insight was innovative and significant. In fact, his insight is an important example of Hammer's (1990) concept of reengineering: rather than trying to automate traditional accounting artifacts, McCarthy advocated a focus on the actual objects in the reality of the business enterprise. Under McCarthy's (1979) approach, a slice of a business reality might show a sale object relating to a customer object, a salesperson object, an inventory object, and the resulting cash receipt object. One would assign attributes to the model to support all information users and not just accountants. This approach was novel because the traditional accounting approach immediately classified the objects in a double-entry format made up of accounting journals and ledgers, which narrowed the primary applicability of the information to accounting users. The traditional accounting approach would have recorded the customer to whom the sale was made only if the sale was made on credit (and, thus, in an accounts receivable subsidiary ledger). One would not have recorded the quantities, unit costs, and selling prices of the inventory items involved in the sale and the salesperson who made the sale, so anyone needing that information would have needed to refer back to the original source documents or to a sales or management information system if one existed. The connection of which cash receipts resulted from which sales would have been tenuous at best.

Dunn and McCarthy (1997) discussed the influence of the concepts of accounting theorists Ijiri (1967, 1975) and Mattesich (1964) on REA and the important distinction between McCarthy's concept of duality requiring causality and traceability rather than simple classification. Dunn and McCarthy also differentiated REA from Sorter's (1969) concept of events accounting. Sorter's concept was a reporting framework that closely resembled cash flow accounting; he did not advocate changing the way one stores transaction details. Indeed, several other papers, such as Goetz (1939) and Everest and Weber (1977), advocated approaches that were more similar to REA than was the events accounting approach in Sorter (1969). Those authors recommended that companies store data that captures multiple dimensions of transactions in various types of database systems. Using three orientations (database, semantic, and structuring), Dunn and McCarthy distinguished between database accounting, semantically modeled accounting, and REA accounting. A database orientation requires data to be stored 1) at their most primitive levels, 2) such that all authorized decision makers have access to it, and 3) such that it may be retrieved in various formats as needed for different purposes. A semantic orientation requires components of the models that reflect only real-world phenomena (rather than double-entry accounting artifacts) as declarative primitives. Such systems may produce artificial constructs as system outputs (e.g., materialized views of the data), but the systems may not include artificial constructs as foundational elements (in other words, declarative data structures). A structuring orientation requires one to repeatedly use an occurrence template as a foundation or accountability infrastructure for the integrated enterprise system. Such pattern-based design facilitates system integration, extension to higher and lower levels of abstraction, and interoperability.

McCarthy (1980) continued to describe how conceptual modeling could be used to design accounting (enterprise) systems. In this work (p. 628), he emphasized four significant limitations of the traditional framework of accounting systems:

1) Its dimensions are generally limited to monetary terms (i.e., dollars) and dates.
2) Its classification schemes are not always appropriate because the chart of accounts often results in data being omitted or classified such that non-accountants do not understand its true nature.
3) Information, generally aggregated to the level of the journal entry dollar amount, is stored at too high of a level of aggregation, which denies other decision makers the ability to use the raw data.
4) The integration with the other functional areas of an enterprise is restricted, which likely results from the above-mentioned limitations; hence, others will maintain the same information, which leads to inconsistent data and information gaps and overlaps.

McCarthy (1980) explained how one could use a conceptual modeling approach to address these limitations by illustrating both declarative (motivated by Chen, 1976) and procedural aspects (motivated by Wong & Mylopoulos, 1977; Chamberlin et al., 1976). McCarthy (1979, 1980) set the stage for McCarthy (1982) in which he defines the REA model's core constructs. If it is not apparent from the discussion above, we make it clear that REA is not about a particular technology nor about a particular conceptual model. Any type of conceptual data model (entity-relationship diagram, UML class diagram, etc.) implemented using any technology (relational, object-oriented, etc.) can represent the REA principles.

As Figure 4 shows, REA is at the intersection of domain-independent theories and concepts from information systems and computer science and those from business, economics, and accounting. REA has its basis in accounting theory, economic theory, database theory, and conceptual modeling. Following McCarthy (1982), scholars subsequently expanded REA into a design theory with additional bases in ontological research, knowledge representations, value chains, and business strategy. As such, it is well suited to serve across multiple fields.

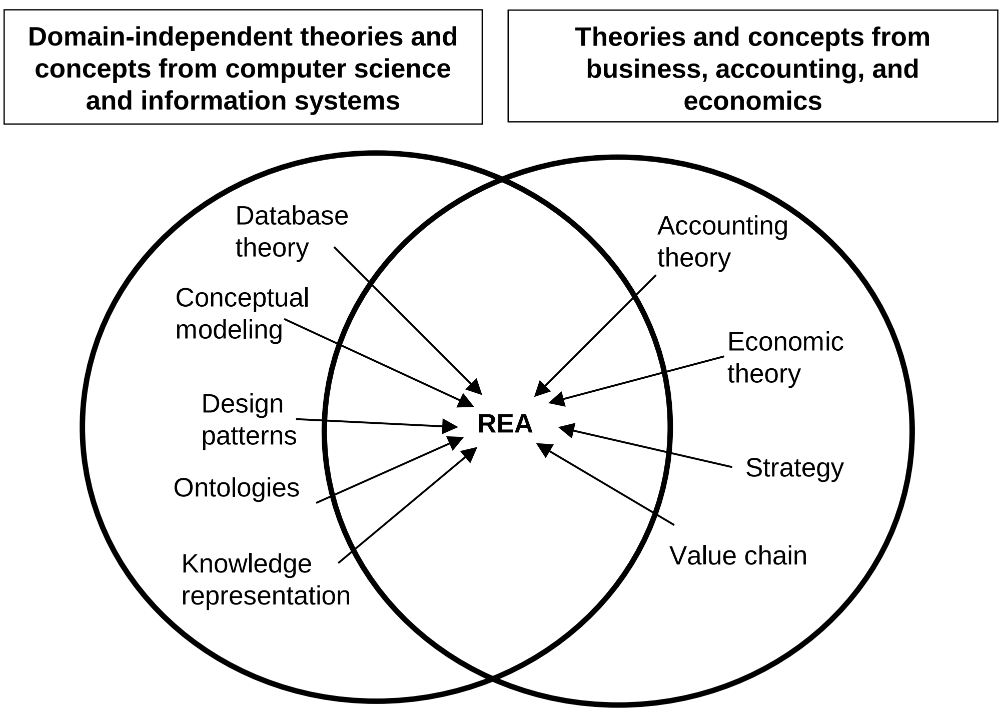

Figure 3. REA as Intersection of Computer Systems/Information Systems and Business Domains

While we focus primarily on the REA literature in this paper, we consider REA's intellectual heritage for additional insights. In order to add richness to Figure 4, in Table A1 (see Appendix), we show a timeline of influential work that allowed REA to mature to its current state. We do not elaborate in great detail on these sources here, but we briefly overview them in Table A1. Along with each paper's details, the table presents each publication's domain area and the manner in which it influenced or demonstrates consistency with REA. This table should help readers understand the importance of REA design theory as a nexus of ideas from the information systems, computer science, economics, and business domains.

In Sections 4 and 5, we summarize REA design science and behavioral science research (March & Smith, 1995; David, Gerard, & McCarthy, 2002; Hevner, March, Park, & Ram, 2004; Dunn & Grabski, 2002). McCarthy (1982) and all of his subsequent research is design science research. Authors necessarily conducted early REA research using the design science methodology. One had to design the framework before it could be implemented, tested, and validated. Design science research that further develops and extends REA continues to this day; however, researchers have subsequently begun to test and validate REA with field studies involving enterprises and with behavioral studies involving individuals.

4 REA Design Science Research

While one can use many different approaches to categorize REA design science research, we divide said research into three categories. First, we discuss the research that further refines the existing constructs in the REA design theory or compares REA constructs to those of other design patterns or ontologies. Second, we describe studies that recommend extensions to REA or that apply REA in new contexts. Third, we discuss research that has created physical implementations or proofs of concepts, some of which also extended REA.

4.1 Refinement and Analysis of REA Constructs

Several studies have suggested refinements to existing REA constructs (which have clarified and, in some cases, expanded their scope) or compared REA to other patterns or ontologies.

Geerts and McCarthy (1997b) acknowledge that not all transaction cycles include simple exchanges and, thus, refine the duality construct. They present a duality pattern taxonomy illustrating eight possible duality representations of varying complexity. While the theoretical ideal is full traceability from the resources involved in economic decrement events to the resources involved in the causally related economic increment events, such full-REA modeling is often hindered because of measurement and aggregation difficulties. Geerts and McCarthy (1997b) describe several of these difficulties and provide object patterns for them.

Borch and Stefansen (2004) conduct a limited ontological analysis focused specifically on the duality axiom defined in Geerts and McCarthy (2000b). Borch and Stefansen (2004) attempt to make the REA ontology fully operational. Their criticism of the REA duality construct centers on three items. First, they argue that the duality axiom is flawed from an operational perspective; that is, it is too simple to model what they refer to as non-exchanges (e.g., pay taxes, make donations). Second, they state the requirement that inflow events must be “paired” with outflow events is ambiguous. Paired implies a one-to-one type-level pattern; however, events is plural and implies a many-to-many type-level pattern. Borch and Stephansen state that the ontology must address cardinalities directly; they believe that relegating cardinalities to a design decision results in an under-specified ontology. An under-specified ontology could result in systems based on the same ontology having significantly different structures and, thus, reduced interoperability. Third, they claim that the REA ontology fails to specify whether duality is a type or an instance. They cite David (1997) who demonstrates duality as a type-level property. Borch and Stefansen note that this approach might result in inflexible systems from a practical perspective. Regardless, they acknowledge the need for consistency because it might not even be possible to integrate systems based on different interpretations.

Hessellund (2006) laments the fact that software developers and domain experts often have difficulty communicating with each other. Domain experts think they are clear as to what they need, and the software developers think they are clear as to what they provide, yet a seemingly unbridgeable gap separates the two. Hessellund (2006) believes that REA can become a ubiquitous language that can bridge the communication gap between business experts and enterprise-system designers. He notes that REA is well suited for ERP systems because it provides a simple but generic organizing principle for enterprise operational data. Whereas traditional modeling schemes are often accounting-specific and, thus, not much use to people from other domains, he applauds REA for having a generic and fine-grained nature that allows accountants and non-accountants to share operational data and create useful reports. However, Hessellund would like to see five improvements made to REA to enable it to become that ubiquitous language.

First, Hessellund (2006) wants the duality construct to be “balanced” to better communicate the timings of the exchanges. We believe that he means to advocate that duality relationships with different multiplicities should be given different labels to better enable interoperability and automated intensional reasoning. For example, because different multiplicity combinations result in different table implementations, the resulting queries needed to calculate the balance of a claim such as accounts receivable are also constructed differently. Had Geerts and McCarthy (1997b) continued their effort to specify eight different duality patterns, they may have satisfied these concerns. Second, Hessellund believes that one needs to define the temporal nature of events (specifically, whether an event is instantaneous or has duration) in the ontology rather than simply capturing it as they currently are with date and time attributes assigned to the event classes. Third, he would like the REA ontology to include a compliance principle to allow one to evaluate how well an instantiated model complies with the REA metamodel (ontology). To meet this need, future research could build on David's (1995) accounting systems characteristics metric. Fourth, Hessellund advocates replacing the term implementation compromise with modeling compromise because the compromise has more to do with the modeling representation than the actual system. Dunn (2012) similarly clarifies compromises by discussing implementation compromises as conceptual-level compromises, logical-level compromises, and physical-implementation compromises. Fifth, Hessellund argues that the REA ontology currently omits roles and future research must determine whether REA should formally incorporate and differentiate the concepts of roles and agents. Along with these five suggestions, Hessellund (2006) also echoes Denna, Cherrington, Andros, and Hollander's (1993) recommendation that REA should feature location as a base object rather than including it as an attribute of an event. He notes that the location construct is especially important in supply chain modeling for concepts such as vendor managed inventory, and he even suggests that transportation (change of location) should be considered as another type of duality (along with exchanges and transformations).

Hruby and Kiehn (2006) extend REA ontology by explicitly considering rights and ownership rights in relation to the REA stock-flow construct that associates events to the resource that is being increased (incremented) or decreased (decremented). The prior REA literature classifies the increment as “inflow” for an exchange process and “produce” for a conversion process and the decrement as “outflow” for an exchange process and “use/consume” for a conversion process. Hruby and Kiehn (2006) suggest refined semantics based on whether resource ownership changes or not. For the exchange process, an increment with a change of ownership would be “buy” and with no change of ownership would be “borrow”; a decrement with a change of ownership would be “sell” and with no change of ownership would be “lend”. For the conversion process, an increment with a change of ownership would be “create” and with no change of ownership would be “produce”; a decrement with a change of ownership would be “consume” and with no change of ownership would be “use”.

Jaquet (2006) examines the concept of transformations as compared to exchanges. She states that REA specifies exchanges well but that it lacks explicit semantics for transformations. She also notes that, when Geerts and McCarthy (2000b) use transformation, they refer to changes of form or substance; however, she notes that transformations can occur in placement, location, or time. Jaquet (2006) further examines the three axioms from Geerts and McCarthy (2000b) and concludes that the axioms are more connected to exchanges and that the transformations also need explicit attention. Furthermore, important properties distinguish exchanges and transformations, such as the number of participating agents (two with exchanges, at least one with transformations), the nature of the duality associations (which connect separate events in an exchange but congruent events in a transformation), temporal relations (exchanges are instantaneous but transformations have durations), and change relations (agent-resource relationships are changed with exchanges, but changes become a property of the transformed resource).

Guan, Cobb, and Levitan (2006) analyze REA using the Bunge-Wand-Weber ontology. They conclude that REA 1) does not allow one to represent a domain's behavioral aspects, 2) lacks ontological completeness, and 3) lacks ontological clarity. However, Geerts and McCarthy (2000b) model behavioral aspects of the domain, and nothing about the REA ontology prohibits behavioral modeling (although we need more research in this area). Guan et al. (2006, p. 3784) also state that the completeness and clarity issues result “from the use of entity-relationship modeling constructs in REA”. However, as we note elsewhere in this paper, REA is independent of the modeling grammar (and recent published REA research uses UML as a modeling grammar). As such, we are unsure as to the severity of Guan et al.'s criticisms relative to REA because they seem to be more related to the entity-relationship model than they are to REA.

Gailly and Poels (2007a) revisit the axioms that Geerts and McCarthy (2000b) propose and argue for the axioms to be specified at the type or instance level because such refinement would improve the operationalization of the REA ontology. Gailly and Poels (2007a), motivated by Guarino (1997), draw a distinction between using REA as an ontology for systems design versus using REA as an ontology at system run-time. In other words, they inquire whether the REA ontology is conceptual or whether it is or can be physical. Gailly and Poels used METHONTOLOGY (Fernández-López, Gómez-Pérez, & Juristo, 1997) to analyze and redesign the REA ontology. They create an REA representation using a UML class diagram and then map that diagram to the Web Ontology Language (OWL) to yield a formal REA specification that can be used at run-time.

Weigand, Andersson, Johannesson, Bergholtz, and Arachchige (2010) apply REA to describe coordination services, such as travel agencies, which match providers who offer services such as childcare, painting, and travel with consumers who need such services. Weigand et al. (2010) suggest that commitments in such a situation are not events but instead are resources because they view the acceptance of orders as economic events and the canceling of reservations as economic events. Alternatively, Weigand et al. (2010) could model the coordination service companies as we would any service providers. For such companies, the resources given up are intangible, whereas the resources received (cash) are tangible. As Dunn (2012) discusses, REA modelers typically substitute a resource type or event type class when the resource is too intangible to measure directly. We see no reason the same approach can't be used for providing coordination services; however, future research could examine this matter more closely.

4.2 Proposed REA Extensions or Applications in New Contexts

In this section, we summarize several research studies in which scholars have either proposed extensions to REA or documented considerations for applying REA in different contexts. Gal and McCarthy (1985) were the first to consider internal controls in an REA system. They point out that, when databases are shared across an organization, risks of inappropriate access to data (i.e., data not commensurate with job functions) arise. Gal and McCarthy used Query-by-Example (Zloof, 1975) to demonstrate how to partition the conceptual schema into several logical views that would map into specific job functions. Furthermore, they advocate storing authorization constraints (over read, write, modify, delete privileges) in a data dictionary. They suggest that future research should examine using internal controls as part of the semantics of the conceptual schema—a research opportunity that still exists today.

Denna, Cherrington, Andros, and Hollander (1993) write about REA for business practitioners as opposed to the research community. They elaborate on the limitations of traditional debit-credit-account (DCA) systems that McCarthy identified in his prior research, and they make a compelling argument for businesses to change to REA systems. They also present a step-by-step approach for analyzing business processes that one wants to implement in REA systems. As part of this approach, they argue that the core pattern should include a location primitive because management sometimes needs to know the location of certain events. While they acknowledge that information about the location of an event can reside in a particular agent or resource associated with an event, they state that, if one cannot determine the location from those other objects, then one must explicitly model it.

Grabski and Marsh (1994) address the need to extend REA to manufacturing information systems and to provide accountability for costs via activity-based costing. They also demonstrate how to use REA in a continuous manufacturing process environment (a natural gas processing plant). They argue that one can often model cost drivers (from an activity-based costing perspective) as external agents associated with the resource consumed in an event. This provides insight for developers of activity-based costing systems about how to model the costing process. Likewise, Denna, Jasperson, Fong, and Middleman (1994) use REA to model conversion-process events. They examine different conversion processes, including commercial fishing, petroleum production, and steel manufacturing. They observe that the external agent has not often been a part of the conversion activities as traditionally characterized. They suggest clarifying the REA model to replace agent with event-specific stewardship and participants (who can be people or machines) that perform the stewardship activity.

With the growing popularity of data warehouses, O'Leary (1999), while stressing the importance of data warehouses for marketing decision making, recognized the opportunity to synthesize REA and data-warehousing concepts. O'Leary notes that data warehouses are largely atheoretical and argues that REA can facilitate a theory-based approach to designing data warehouses. As a result, he creates a schema called REAL-D; the REA extensions L and D are for location and data warehouse, which become specific dimensions to facilitate data aggregation. One can include these data aggregations as part of the REA theory in Geerts and McCarthy (2002); however, the difference is that it would be part of the central enterprise system as opposed to a separate data warehouse. One does not need data warehouses when the memory and processing capacities allow one to directly query and manipulate operational data.

Geerts and McCarthy (1997a, 1999) use a three-level architecture based on abstraction of exchange patterns to extend REA research. At the highest level of abstraction is the enterprise value chain. Geerts and McCarthy (1999) describe the enterprise value chain level as a fundamental script involving economic exchanges in which an entrepreneur obtains some capital/financing, purchases production inputs, produces some output, sells the output to a customer, and settles the financing. According to Geerts and McCarthy (1999, p. 89), these exchanges form “a chain of economic exchanges…each time giving up an economic resource (perhaps money) in return for another resource of greater value. Value is defined as a deliverable portfolio of product or service attributes attractive to the firm’s ultimate customers.”. The second level of the architecture takes each exchange in the enterprise value chain and maps it to an REA pattern. The third level of the architecture takes the key events in the exchange and provides the tasks or workflow steps that comprise the event. Geerts and McCarthy (1997a) provide an in-depth example of this architecture. This architecture also forms the conceptual basis for a CASE tool/framework called the framework for REA accounting (FREACC) (Geerts & McCarthy, 1997a, p. 105). Geerts and McCarthy (1997a) also contribute to REA research by asserting REA as a design pattern as evidenced by analytical comparisons of REA with Coad’s (1995) patterns, Jacobson’s (1992), and Jacobson, Ericsson, and Jacobson’s (1994) use-case analysis and Gamma, Helm, Johnson, and Vlissides’ (1995) design patterns. Also notable is Geerts and McCarthy’s (1999) characterization of REA as an ontology.

Geerts and McCarthy’s (2000a) work on REA and augmented intensional reasoning is innovative. REA is not used solely for systems design; rather, REA, or more specifically, REA knowledge in the form of both declarations and procedures, is explicitly embedded in the information system. For example, an information system can store a declarative definition for a claim: “A claim with an outside agent exists when there is a flow of resources with that agent without the full set of corresponding instances of a dual flow” (Geerts & McCarthy, 2000a p. 136). Geerts and McCarthy also provide a Prolog implementation to show how claim materialization works (in the context of a broader system called conceptualizing REA systems (CREASY)).

4.3 REA Proofs of Concept

After McCarthy (1982) published the REA accounting model, Gal and McCarthy (1983) were the first to physically implement the concepts. Gal and McCarthy mapped a business object system to an entity-relationship model and mapped the entity-relationship model to a CODSAYL model. This work served as a proof of concept and reinforced the notion that one can ignore traditional accounting artifacts in a system’s data structure design and yet still have available the necessary data to materialize account balances and financial reports, including financial statements. Gal and McCarthy (1986) extend their 1983 work into a relational database implementation again using Query-by-Example. Consistent with all of McCarthy’s research, Gal and McCarthy do not use or need accounting artifacts such as a general ledger in the design process. However, Gal and McCarthy (1986) portray the general ledger as a view or output of the system (as opposed to a data structure in the system); Gal and McCarthy (1986) overview the hierarchical procedures used to materialize a general ledger from raw transactional data.

While Gal and McCarthy’s (1986) prototype used Query-by-Example and a retail business as the domain of discourse, Denna and McCarthy (1987) built a prototype of a manufacturing business using Knowledge Manager (Holsapple & Whinston, 1984), which augments the relational database management system with decision support capabilities such as spreadsheets and graphics. This prototype advanced the procedures used to materialize a general ledger from transactional data. A foundational REA database supported decision support capabilities of the prototype system, which external databases and statistical analysis supplemented. The ideas in this paper are clearly relevant today with ERP software supplemented by external data available via the Internet, which foreshadowed the data analytics and presentation capabilities inherent in current enterprise systems.

Other early REA research looked to provide additional proof of concept for the REA design theory and used computer-aided software engineering (CASE) concepts and tools. Chen, McLeod, and O’Leary (1995, 1998) use REA in a prototype CASE tool called REAtool. REAtool supports schema evolution, a process in which an inputted schema is converted to an REA-compliant schema. Chen et al. developed REAtool using REA domain knowledge and embedding that knowledge along with evolution heuristics in a prototype system a schema evolution administration tool (SEAtool). REAtool and SEAtool operate on an object-oriented database management system. Chen et al. (1998) say future research has an opportunity incorporate additional domain knowledge into REAtool based on firm type (e.g., service vs. manufacturing) because REA domain knowledge is at a more general level.

Geerts, McCarthy, and Rockwell (1996) summarize additional REA CASE tool research. One such tool, called REACH (described more fully in Rockwell and McCarthy (1999)), incorporates three types of knowledge (REA knowledge, reconstructive expertise, and implementation compromise heuristics) to aid in view modeling and view integration. Although the authors do not use this specific terminology, the REA knowledge in the CASE tool is essentially the REA design theory. It is used to guide users in developing models. The reconstructive expertise embedded in REACH provides information needed to generate traditional accounting views of data with things such as templates for the general ledger chart of accounts for particular industries. The implementation compromise heuristics are a key part of REACH because the theoretically ideal REA data structure is typically constrained in practical implementations. The constraints may, for example, arise from technological or measurement limitations. Although part of REACH is conceptual and not fully implemented, Rockwell (1992) and Rockwell and McCarthy (1999) implemented a prototype in a knowledge-based system called REA view integration with expertise from written sources (REA VIEWS). Going beyond CASE tools, Murthy and Wiggins (2004) developed an object-oriented extension to REA. They use UML notation and highlight the advantages of object-oriented design: integrating data and processes in one model and the UML modeling formalism that maps directly into object-oriented environments.

Gailly and Poels (2007b) present complementary research to extend REA from a business domain ontology to a business modeling ontology. Gailly and Poels add the REA axioms that Geerts and McCarthy (2000b) propose to their redesigned conceptual representation, which represents another opportunity to formalize REA constructs and axioms. Research opportunities exist to try and validate this recent line of research. Gailly, Laurier, and Poels (2008) also use the METHONTOLOGY approach to extend REA and provide a proof-of-concept implementation using OWL and UML. They advocate using REA ontology-driven business modeling and demonstrate, using Protégé, how to use a machine-readable representation of the REA ontology to model a simple process.

Sedbrook and Newmark (2008) also tried to unite semantic Web technologies, including OWL, with REA. They used their REA-based OWL ontology to model policies for a distributed e-commerce partnership that sold barely used cars. The partnership included several types of vehicle suppliers and marketing partners. They developed a prototype using REA and OWL and applied Semantic Web Rule Language (SWRL) to integrate the views of the various business partners. They found that the combination of REA and OWL enforced consistency in integration and reasoning across ontologies, and that SWRL provides a good representation of partner exchanges.

Geerts and McCarthy (2000b) and (2002) build on the original core pattern primitives (McCarthy, 1982) and make significant design science contributions by presenting new primitives (types, commitments, association, custody, reserves, executes, and reciprocal) and providing a formal ontological analysis of the extended REA architecture. While the original REA primitives provide an accountability infrastructure, the extended primitives provide a policy infrastructure. The accountability infrastructure allows REA models to represent things that have occurred (i.e., to provide a historical record of enterprise activity). The policy infrastructure uses typification to extend the original REA model into the realm of what could be or what should be as enterprise policies. Consistent with the REA ontology, Allen and March (2003, 2006) advocate that events should be included in enterprise ontologies because time is an integral part of business systems' semantics. They recommend preserving temporal information such as the time events that occur and recording the time events (and state history if required).

Hruby (2003) extends the notion of REA as a design pattern (Geerts & McCarthy, 1997a) to include several behavioral patterns to support business system modeling. As a result, Hruby formalizes several business patterns such as business relationship, business transaction, contract, role, account, due date, address, and classification. The entire set of business patterns aggregates to a pattern map. Hruby (2005) uses REA ontological categories as a metamodel and illustrates how one could instantiate the metamodel with application objects. He argues that user-specified requirements make it difficult for an ontology to describe functionality differences across systems in a domain. However, one could resolve this problem by having an object dimension related to the ontological requirements and an aspect dimension that would account for user-specified requirements.

Batra and Sin (2008) argue that traditional REA models are too data oriented and not oriented towards dynamic behavior. They use UML sequence diagrams to suggest how one could extend REA to represent dynamic behavior. Their UML sequence diagrams include a generic diagram and instantiated diagrams for a sales order scenario, an invoice scenario, a collection scenario, and a work-in-process scenario. Similarly, Vymetal, Hunka, Hucka, and Kaski (2010) create a dynamic REA form to depict the workflow underlying the value chain-level REA models and operational-level REA models with state, activity, and sequence diagrams. They recommend using the REA business model to identify resource, agents, and so forth. They suggest one start to create an activity diagram by making a swim lane for each agent and illustrating resources in the border lines between two adjacent agents. They further recommend using the activity diagram to produce state and sequence diagrams to describe different views. However, they note that they do not include any iterations or cycles in their models and that they use a simple model that omits other processes. They encourage future research to extend their approach to more complicated scenarios. While we agree that one can represent the workflow underlying REA business models with UML activity, state, and sequence diagrams, we are concerned that there is no standard script to which the underlying workflow conforms for all companies.

Sedbrook (2010) developed and field tested a maintenance methodology for the REA enterprise ontology. He used a methodology called ready, intermingle and accept (RIA) with an OWL formalization to automate domain change maintenance for a semantic wiki. The ready phase includes classifying domain documents in RDF and updating the domain classes, taxonomies, and relations. The intermingle phase involves classifying domain properties in REA property structures and inferring REA OWL classifications and properties. The accept phase includes transforming the REA ontology to the Wiki RDF and updating the wiki pages, properties, and categories. He field tested his methodology by mapping the Association to Advance Collegiate Schools of Business' assurance of learning standards to the REA ontology and creating reusable SPARQL queries to extract content for wiki page creation and maintenance. The RIA methodology, with REA embedded in its intermingle phase, enabled the query language to automatically update the wiki and improved consistency and coherence.

Mayrhofer (2010) examines reference modeling design approaches and, based on inter-organizational models, creates a prototype implementation of inter-organizational modeling. Mayrhofer states that this approach will result in efficiency gains in the modeling process and also result in higher-quality models. Mayrhofer develops a three-level model (value perspective, process flow perspective, and execution perspective) with the REA ontology used for the value perspective (along with e^3value). The value perspective captures the economic resource exchange between business partners. The process-flow perspective provides the business process models and is based on the UN/CEFACT modeling methodology. Mayrhofer then translates the defined models into artifacts that one can use on the IT layer (e.g., APIs). Mayrhofer calls this approach business semantics on top of process technology (BSpot). He reports a problem with the use of BSpot tool: if an organization wanted to use and modify an existing process model, even if the change is minor, the organization needed to completely start again from the beginning without leveraging the existing technology. As a solution to this problem, Mayrhofer proposes that using reference models (e.g., REA) will spur system designers to re-use and recreate modified business and process models.

Sonnenberg, Huemer, Hofreiter, Mayrhofer, and Braccini (2011) develop a domain-specific modeling language (DSL) with graphical syntax to communicate various elements of the REA ontology for business models so that traditional modeling tools such as entity-relationship diagrams or UML class diagrams need not communicate the REA patterns. In addition to creating the DSL, they implemented it using Microsoft Visual Studio 2010 Visualization and Modeling SDK as a proof of concept. The DSL in their research includes the core REA model and value chain; they anticipate extending the DSL to include the ontology's extensions such as commitments, typification, and policy infrastructure and enabling one to automatically design the underlying database. Further research may demonstrate whether REA represented in a graphical syntax such as the DSL is superior to REA represented in non-specific language syntaxes such as UML.

Sedbrook (2012) also advocates using a DSL. He built a prototype with a meta-model that specifies REA modeling components and a visual interface to design both operational- and policy-level REA models. The prototype, called tracing enterprise architecture (TEA), also included code-generation templates to allow one to automatically transform design models into executable code. As Sonnenberg et al. (2011) did, he implemented TEA in the DSL framework using Microsoft Visual Studio. He used TEA to define REA primitives and commitments and to enforce REA's axioms for duality, stock flows, participation, and reciprocity. Modelers who tried to form a graphical connection that would violate REA semantics were either restricted from creating the connection or were warned with a descriptive message.

Laurier and Poels (2013) use the REA ontology as a basis to develop a simulation across multiple organizations in a supply chain. Designers perform this simulation at various abstraction levels (value system, supply chain, value chain, and business process). They adapt the original REA axioms (Geerts &

McCarthy, 2000b) to the supply-chain environment. They developed the simulation to demonstrate the benefits of using the REA ontology for discrete-event simulations of value systems. The unique contribution of this research is that it uses the REA ontology across the multiple supply-chain parties (e.g., customer, distributor, manufacturer). The research also demonstrates the re-use of a given supply-chain entity's simulation for other entities or, alternatively, as a standalone model with minor modifications. The authors claim that this re-use in dynamic environments provides a significant advantage over other simulation approaches.

Hunka and Zacek (2014) evaluate the REA ontology from the perspective of design and engineering methodology for organizations (DEMO). They analyze the REA ontology to determine whether it has any deficiencies to provide a guideline for improving it. They also investigate whether they can improve DEMO. They claim that the REA core pattern captures only past and present events (i.e., that "future" events are out of REA's scope), which was never the situation with the REA framework (which evolved into the ontology). Both the framework and ontology allow future events, such as orders and commitments. The researchers also observe that the REA ontology should identify and declare business transaction states and state transitions directly instead of creating the REA state machine with business transaction phases. The authors also state that the REA ontology is "missing a vigorous theory and methodology" (p. 74). However, McCarthy (1982) took great care to ensure the REA elements faithfully represented concepts in accounting theory as Mattesich (1964) and Ijiri (1967, 1975) have documented.

4.4 Summary of REA Design Science Research

REA transformed from a framework to a design theory for inter-organizational exchange transactions with prototypes and implementations in various types of semantic Web technologies. Scholars have expanded REA at the business-process level to incorporate base constructs needed for management planning and control functions, and some scholars even expanded REA to include value chains, value systems, and task-level concepts (recall Figure 3). REA has gained international recognition and acceptance as a viable enterprise ontology. While earlier design science research on REA focused on the US, much recent work has additionally occurred in Europe and Asia. As such, the research community has benefited REA by calling more attention to the importance of logically formalizing the ontology and fully operationalizing it in running systems (state machines). The notions that the ontology must be made complete and that one should not use implementation compromise to justify incompleteness comes primarily from outside the US. Non-U.S. researchers have also strongly advocated the usefulness of extending REA from the trading-partner view to an independent view. Clearly, we need more research in these areas.

We conclude this section cautioning researchers about some common misunderstandings regarding REA. One we previously mention is that researchers need to understand that REA concepts exist independent from any chosen modeling formalism. Some studies feature a misconception that one can depict REA only as an entity-relationship model. Researchers need to take care not to make inappropriate conclusions about REA when those conclusions may more appropriately be directed at the modeling formalism. Other research reveals a misconception that REA diagrams must always follow the physical layout with resources on the left, events in the center, and agents on the right. Rarely can one organize an entire company's REA model so cleanly. We do not claim that physical layout doesn't matter; indeed, some of the behavioral research we review in Section 5 demonstrates performance differences with different layouts. However, if REA's basic elements are present in a diagram that is not laid out in the left-to-right fashion, it is inaccurate to say that is not a REA model. We correct such misunderstandings and provide clarifications to allow future research to build on the extant design science research on REA. In Section 5, we present the behavioral research that has investigated how REA would benefit the users of systems based on REA.

5 Behavioral REA Research

Dunn and McCarthy (1997) encouraged behavioral researchers to study semantic models of accounting phenomena because of the lack of empirical work in this area. Five years later, Dunn and Grabski (2002) noted a continuing dearth of behavioral research on REA and other semantic accounting models. Dunn and Grabski (2002) summarize the findings of semantic modeling studies, categorize those studies according to whether they encompassed surface- or deep-level semantic structures, and summarize a variety of cognitive psychology and information systems theories that one could use or combine to frame future REA behavioral research.

David, Dunn, McCarthy, and Poston (1999) establish a research pyramid for classifying research on semantically modeled accounting systems and, thereby, identify new research opportunities (see Figure 5).

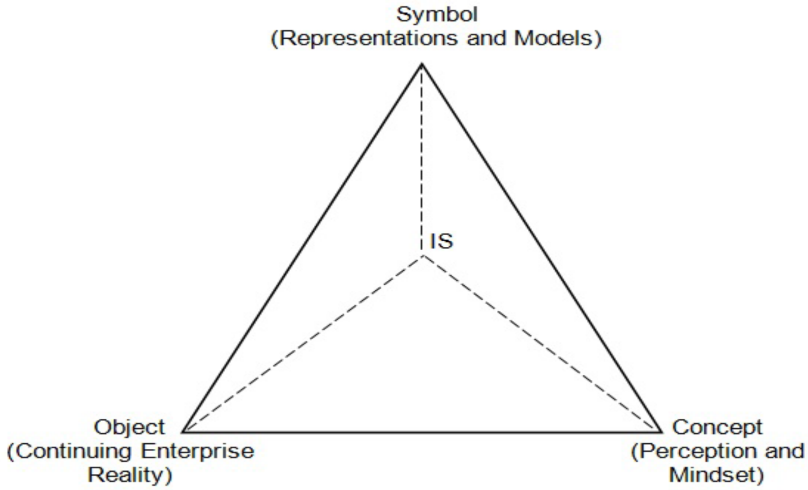

Figure 4. The Research Pyramid (Adapted from David et al., 1999)

The pyramid's three base points come from Sowa's "meaning triangle" (Sowa, 1999) and illustrate that humans perceive real-world objects (e.g., real things existing in the day-to-day operations of a company) as concepts (e.g., perceptions, mindsets, schemas, and mental models) and represent them as symbols in linguistic, paper, or electronic form for communication with other humans (e.g., source documents, conceptual models, graphs, and tables). The information systems corner that extends the triangle into a pyramid illustrates that one can implement these symbol sets (as representations of perceived objects) as working information systems. Behavioral information systems research often involves the information system-concept edge in isolation (e.g., what effect do systems have on user perceptions or what effect do system designers' mental models have on the resulting system). Incomplete theory regarding cognitive processes and individual differences often makes it impossible to replicate the results of a study with similar participants and similar systems. Studies encompassing the pyramid's IS-symbol-concept face rather than just its IS-concept edge may be useful for developing human-computer interaction theory. One may identify consistencies in symbol sets across systems and compile the results of many studies to reveal patterns from which one can propose and test theory. David, Dunn, and McCarthy (1999) emphasize the need to compare enterprise systems at the symbol-set level rather than at the implementation level and further propose that REA is a viable candidate for an ideal symbol set against which one can evaluate other ERP symbol sets.

With any system, one can categorize behavior into three basic types: behavior in designing and implementing the system, behavior in using the system, and behavior in managing, monitoring, or auditing the system. We categorize the behavioral REA accounting research according to these three types. We also indicate which edges or faces of the research pyramid each paper includes to give researchers some insight about how they might expand the existing behavioral REA literature with future studies.

5.1 REA System Design and Implementation

System design and implementation comprises requirements analysis, conceptual modeling, conversion of conceptual models to logical database models, and physical implementation of the system. Relatively little behavioral REA research has been done in this area, but what has been done demonstrates the potential for REA to facilitate the system design and implementation process. Poels, Maes, Gailly, and Paemeleire (2011) found REA to facilitate requirements analysis. Dunn and Grabski (1998) and Gerard (2005) found REA to facilitate conceptual modeling. All three studies in this category examine the research pyramid's concept-symbol edge because they all measure users' schema (concepts) and study behavior with conceptual models (symbols). To our knowledge, no studies have observed whether REA helps with converting conceptual to logical models or with the physical system implementation. As such, researchers could conduct such research.

Poels, Maes, Gailly, and Paemeleire (2011) examine the requirements engineering process of designing enterprise systems. They conducted an experiment in which participants answered comprehension questions after reviewing REA and non-REA diagrams. They found higher accuracy with REA, no significant difference in efficiency, and higher overall efficacy with REA diagrams. They concluded that using the REA core pattern increases the conceptual schemas' pragmatic quality, which means that the users' interpretation of the domain semantics that the schemas conveyed was closer to what the system analysts intended. The improved analyst-user communication helps ensure the success of the requirements-validation task. Users that accurately understand the conceptual schemas can better detect incompleteness and invalidity of requirements specifications; thus, the higher a REA-modeled schema's pragmatic quality, the higher the end product's quality (the enterprise system). This research supports Hessellund's (2006) belief that REA can bridge the communication gap between business experts and enterprise-system designers.

Dunn and Grabski (1998) hypothesize that people who can better separate constructs from their context (i.e., field-independent people) create better conceptual models than do people who have difficulty in recognizing the same construct in different contexts (i.e., field-dependent people). Using entity-relationship diagrams to depict the ontological constructs, the authors tested this hypothesis and found that their field-independent and field-dependent participants performed equally well on non-conceptual modeling tasks, which evidences that one group was not simply smarter than the other. However, the field-independent participants created conceptual models that were more accurate than those of the field-dependent participants.

Gerard (2005) analyzes user schema as a determinant of conceptual modeling performance by combining Weber's (1996) method of using free recall of conceptual models with Chase and Simon's (1973) method of using actual versus random positioning of chessboard game pieces. Chase and Simon compared expert and novice free recall performance and found that expertise level did not significantly relate to free recall performance when these groups of people recalled random chessboard positions; however, the experts recalled significantly more than the novices recalled when using chessboard positions taken from actual chess games. Chase and Simon inferred that people develop schemas when they develop expertise. Thus, chess expertise does not simply mean superior memorization ability but an ability to structure the knowledge in memory (i.e., create a schema) such that the knowledge could be stored and retrieved in larger chunks. Gerard measured users' schemas to determine the extent to which they were consistent with McCarthy's (1982) REA model. To measure the user schemas, Gerard administered free recall tests for an REA model and for a randomized model that contained the same entities and the same number of relationships as the REA model but without the underlying REA structure (e.g., without stockflow relationships). To elaborate, in an REA model, we expect specific relationships, including duality between related economic give and take events (e.g., sale and cash receipt), stockflow between economic events and resources (e.g., sale and inventory), and participation between events and the agents involved in those events (e.g., sale and salesperson or sale and customer). We do not expect relationships between unrelated entities (e.g., sale and cash disbursement or sale and accounts payable clerk). The REA and randomized diagrams in Gerard (2005) were identical in spatial orientation and number of entity symbols (rectangles) and relationship symbols (diamonds). However, the REA diagram included only relationships that the REA ontology prescribes, whereas the randomized diagram contained only nonsensical relationships. This approach was similar to how Chase and Simon (1973) measured chess experts' schemas when they freely recalled chessboards containing pieces in positions that would actually occur in a chess game and chessboards containing pieces in positions that would not occur in a chess game. The chess experts recalled significantly more of the actual game chessboards than the novices did, but the former could not recall any more of the nonsense chessboards than the novices did. Similarly, Gerard found that those with well-developed REA schemas could recall significantly more of the REA diagram than those without well-developed REA schemas, whereas recall performance on the non-REA diagram was similar for all users. Participants in Gerard's (2005) study also designed an REA model for a revenue process of a company and Gerard rated the accuracy of the participants' designs. Gerard's data analysis showed that participants with well-developed REA schemas outperformed participants without well-developed REA schemas when those participants completed the REA modeling task.

5.2 REA System Use

Studies in this category focus on system use. Typical measures of interest in this category include users' ability to comprehend a system and their ability to retrieve information from a system. Therefore, we subdivide this category into those focusing primarily on comprehension versus those focusing primarily on information retrieval. Often, to increase an experiment's internal validity, researchers substitute the conceptual model (symbol) underlying a system for the system itself. Therefore, only Dunn (1994) and Allen and March (2006) incorporate the entire IS-symbol-concept face of the research pyramid; the remainder include the symbol-concept edge.

5.2.1 System Comprehension

Dunn and Grabski (2000) hypothesize that people will perceive the REA accounting model as more semantically expressive (i.e., more completely representing reality) than the traditional DCA accounting model. They also hypothesize that higher perceived semantic expressiveness is associated with higher task accuracy. Participants in their study completed four information retrieval tasks—two with REA-based system documentation and two with DCA-based system documentation. They randomized task and documentation order and controlled for field dependence, cognitive fit, and accounting knowledge. After completing two tasks with one system, participants answered Likert scale questions about the system's semantic expressiveness. Participants then completed two tasks with the alternative system and answered Likert scale questions about that system's semantic expressiveness. Participants found REA to be significantly more semantically expressive than DCA. Further, participants completed tasks more accurately with whichever system they perceived as more semantically expressive.

Dunn and Gerard (2001) compare auditor search, recognition, and inference using REA models in diagrammatic and linguistic formats. Participants who used the diagrammatic format completed tasks faster, perceived the model as easier, and were more satisfied than were participants who used the linguistic format; however, Dunn and Gerard observed no difference in the study participants' accuracy. The latter finding is intuitively appealing because both diagrammatic and linguistic representations are based on the same REA model. This research demonstrated that while one presentation form of the REA model may result in faster results, the accuracy is invariant of the presentation format.

Maes and Poels (2007) evaluate the quality of conceptual modeling scripts (defined as the product obtained by applying the process of conceptual modeling in an actual enterprise) created with the REA ontology to study the effects of perceived semantic quality and perceived ease of understanding on user satisfaction and perceived usefulness. In one experiment, they created two REA models, one with a higher degree of accuracy and, thus, higher semantic quality. The results of that experiment revealed significant effects of perceived ease of understanding and perceived semantic quality on both perceived usefulness and on user satisfaction. They also observed a significant effect of perceived usefulness on user satisfaction. In another experiment, they created two REA models, one of which was not displayed according to the common display layout that many REA follow, with resources on the left, events in the middle, and agents on the right. Both experiments had consistent results.

Poels (2011) provided participants with low, medium, and high exposure to REA patterns with two different diagrams to answer comprehension questions. Both diagrams represented the same domain and included the same overall REA constructs but differed in physical layout and the way in which many-to-many associations were displayed. While Poels labeled them as REA and non-REA diagrams, the meaning Poels ascribes to the term REA *pattern* is the physical layout in a diagram of resources on the left, events in the center, and agents on the right. Thus, one diagram followed that layout and the other did not, though both included the same classes and associations. The only other difference in the informationally equivalent diagrams was that, in the non-patterned layout diagram, the many-to-many associations were objectified (i.e., the author converted corresponding association class into a class with two one-to-many associations to the related classes and, thus, mixed conceptual and logical level modeling). The REA patterned diagram users scored significantly higher on a comprehension task than did users of the non-patterned diagram. Post hoc tests revealed a significant difference for levels of training: users with medium and high training comprehended the diagram better than users with low training. Results indicated no time difference and a weak effect for perceived ease of use/interpretation. Because both diagrams were in essence REA diagrams but with different physical layouts, this research demonstrates that the format in which one presents information is important.

5.2.2 Information Retrieval

Dunn (1994) programmed two user interfaces to a set of 33 database tables based on an REA conceptual model. One interface was an abstraction hierarchy in which users could select a transaction cycle from the enterprise value chain to drill down to the conceptual model for that cycle and drill down further to any entity or relationship and then to the corresponding tables. The other interface included no model, and table names were abstract (table 1, table 2, and so forth). Users of that interface simply clicked “next table” and “previous table” buttons to scroll between tables. Dunn hypothesized that the drill-down interface would aid users in finding the information they sought. Instead, users with the drill-down interface were less accurate than those who simply scrolled the tables, and perceived ease of use was equivalent between the two interfaces. However, we don’t know the participants’ level of REA understanding and how it affected the counterintuitive finding of lower accuracy for the drill-down user group.

Dunn and Grabski (2001) used REA and DCA systems to deepen our understanding of cognitive fit by extending it to the domain of accounting models. They tried to identify what may be occurring inside the mental representation box of Vessey’s (1991) cognitive fit model. Participants performed various information-retrieval tasks with REA and DCA accounting system models, and their performance revealed that localization (how proximally located the needed information is) in graphic presentations is an important element of cognitive fit. REA system users, regardless of their level of expertise (novice or experienced), were able to perform at least as well as (or better than) experienced DCA system users when presented with information localized on the conceptual models. When the data were not localized on the graphical models, then experience mattered.

Allen and March (2006) hypothesize that query accuracy, confidence, and proficiency in self-assessment (i.e., prediction of accuracy) is better for users of conceptual models that explicitly represent events as entities as compared to users of conceptual models that only allow states to be represented as entities. They used REA as the basis for their conceptual model and logical database that explicitly represents events as entities. They created the state-based conceptual and logical models based on Dey, Barron, and Storey (1995) and Teorey, Yang, and Fry (1986). Allen and March (2006) also created artifact-based conceptual and logical models that mixed the state-based and event-based constructs. They assigned study participants to one of the three models (REA, state-based, and artifact-based) and asked participants to perform query tasks. Participants’ query accuracy was not significantly different across the three models, nor was confidence. However, participants using event-based models expressed confidence that better predicted the accuracy of their queries than did subjects using state-based models.

Buder and Felden (2012) evaluated the existing business modeling methods, REA model, and the e^3value model. They found high user understanding with both models not only for the business models but also for business processes. Because e^3value requires fewer concepts to describe the exchange of resources and rights than REA and because REA does not have its own modeling language, they expected to find e^3value as more effective and more efficient than REA. They also expected e^3value to give more detailed information to guide business processes than REA. They gave inexperienced users 90 minutes of training that included both REA and e^3value. In another session, users completed two case studies (one that the authors created for e^3value and one that they created for REA) and answered comprehension questions with REA and with e^3value with order counterbalanced across groups. The e^3value users were more accurate than REA users on only the case that the authors created for e^3value, and the authors found no difference in efficiency. The authors then combined the two case results and concluded that e^3value was overall easier and more efficient to use.

Dunn, Gerard, and Grabski (2013) combine and extend the work done in Dunn and Grabski (2001) and Gerard (2005) to learn more about the effects of cognitive fit and of user schemas on performance. They used free recall tests similar to the one Gerard (2005) used to measure the extent to which users had developed REA and DCA schemas. Users performed REA-facilitated and DCA-facilitated tasks with REA or DCA accounting system structures. The authors demonstrate that one can obviate the long-known relationship between user schema and task performance by a lack of cognitive fit between a representation and the task one needs to complete. As in Dunn and Grabski (2001), Dunn et al. (2013) did not seek to prove anything about REA or DCA but to take advantage of the rich experimental context provided by the alternative models of accounting phenomena to learn more about cognitive fit and user schemas.

5.3 REA System Management, Monitoring, and Audit

We found only two studies in this category, both of which investigate how system auditors interact with REA models. The finding of no studies on REA system management and monitoring indicates a pressing need for research in those areas. certainly, two studies are not comprehensive, so we need more studies on REA system audit. Both studies in this category used the system’s underlying symbols to substitute for the actual system. As such, both studies focus on the research pyramid’s symbol-concept edge.

Dunn, Gerard, and Grabski (2011) demonstrate the importance of the physical layout of REA models although not in the same R-E-A layout that Poels et al. (2011) prescribes. Dunn et al. (2011) propose the theory of diagrammatic attention management that says diagrams need to include some form of attention direction mechanism to aid performance. They tested their theory with REA models presented in four formats: aggregate diagrammatic (a full transaction cycle on a page in entity-relationship diagram format), disaggregate diagrammatic (the same entities and relationships as in the aggregate diagram but presented as a series of binary relationships), aggregate sentential (the same full transaction cycle on a page but in a linguistic representation called BNF grammar), and disaggregate sentential (each relationship presented in binary BNF grammar format). Although most entity-relationship models in practice are portrayed in the aggregate diagrammatic format, Dunn et al. (2011) found users of this format to be the least accurate on a cardinality validation task compared to users of the other three formats, which indicates a need for attention direction for system auditors who use such diagrams.

Dunn, Gerard, Grabski, and Boss (2013) used the REA context to explore optional and mandatory participation in conceptual model relationships—constructs that IS ontology research (e.g, Bodart, Patel, Sim, & Weber, 2001; Bowen, O'Farrell, & Rohde, 2006; Gemino & Wand, 2005; Wand & Weber, 2002), has studied. Dunn et al. (2013) demonstrated better error identification performance when the semantics underlying a conceptual model represent mandatory participation (consistent with the extant ontology research) and provide evidence that the asymmetry in favor of optional participation observed in some prior research was likely caused by the framing of the task prompts rather than the optionality of the participation. The results also indicate that an asymmetry in favor of flexible rather than restrictive maximum participation occurs, which is consistent with psychology research that has demonstrated that people desire to keep options open (Ariely, 2008).

5.4 Summary of REA Behavioral Research

Several basic findings emerge from our analysis of REA behavioral research. Consistent with Hessellund (2006), REA does appear to have the potential to serve as a bridge for the communication gap between domain experts and system developers. REA results in higher levels of pragmatic quality; that is, users of the REA diagrams can better understand what systems analysts try to convey in the conceptual models and the users can better identify inconsistencies and errors. Future research needs to examine whether even better performance would result if REA were more fully developed along the dimensions that Hessellund (2006) suggests. Additionally, findings across multiple studies suggest that REA models with higher levels of semantic quality result in increased perceived ease of use and higher user satisfaction. Individuals identified as more field independent performed better with REA systems than those identified as less field independent. Because almost all of the REA behavioral research has built on tasks performed by undergraduate students, we do not know whether findings hold true for professionals.

Several noteworthy findings have emerged regarding the representation of the REA conceptual model. First, REA presentation format (diagrammatic or linguistic) does not impact performance. However, if REA conceptual model users receive training to expect a basic representation pattern (resources on the left, events in the center, and agents on the right), then, when that layout pattern was not readily apparent, users performed worse than those who received a data model consistent with the physical layout. This finding shows that, consistent with research from many other domains, people follow patterns of use and, when that pattern is “broken”, they perform worse. Other research has found that users of REA models that were not presented in the physical layout (but were informationally equivalent) performed better at information-retrieval tasks than users presented with DCA models.

Other REA-related research has extended our understanding of cognitive fit. This research demonstrates that localizing information in REA data models is an important factor to improve information-retrieval performance. Additional research using REA data models that builds on this finding has identified the need for some type of attention-directing mechanism when one uses conceptual models. Individuals, when presented conceptual models in aggregate diagram format, perform worse on validation tasks than users who were presented the same information in binary diagram form or users who were presented the same information in sentential form (either aggregated or binary). Because users experienced the same volume of information and because the sentential approach has an inherent attention-directing mechanism, the only explanation for the difference in performance is the lack of an attention-directing mechanism for the individuals who viewed the aggregated diagrams. This finding identifies a potential limitation with the way one traditionally evaluates data models and calls into question whether system auditors can improve their reviewing of data models and, thereby, reduce development and maintenance costs. Consistent with ontology research, researchers basing tasks on the REA model found that system auditors more accurately identified errors in multiplicities when the associations had mandatory participation rather than optional minimum participation. Interestingly, the research found that system auditors more accurately identified errors in multiplicities when the associations had optional rather than restrictive maximum participation. These asymmetries and the conflict between them need further investigation.

Analyzing the behavioral research reveals the wealth of opportunity for additional research. We identified only two studies on managing, monitoring, and auditing REA systems, which reveals an opportunity for future research. Even in the system-design and system-use areas, relatively little behavioral REA research exists. Most REA research has been design science oriented, which is not surprising given that researchers had to design REA before behavioral science research could evaluate it.

REA has evolved from an accounting framework to a design theory. It now includes policy-level, commitment-level, and operational-level models. REA includes the value system (inter-organizational REA models), the value chain (the enterprise's script for doing business), the business-process level (transaction cycles), and the workflow level. As such, REA has grown into a robust design theory. Behavioral research has found that REA models improve communication between analysts and users and that REA conceptual models have resulted in a higher perceived ease of use than other conceptual models. Despite the identified need for much additional behavioral research, REA has had an important impact on the research community. However, has REA had any impact on the professional community? We address this question in Section 6.

6 Influence of REA on Practice

REA has had an impact on practice because standard setters and software vendors have incorporated REA into standards and software. REA serves as the basis for an ISO standard, and researchers have found that ERP systems are consistent with REA. Finally, REA serves as a foundation for ERP systems. The semantics contained in the expanded REA model facilitate information exchange between trading partners and provide a needed foundation for ERP systems. In this section, we first present research identifying REA's impact on current enterprise systems. Second, we discuss an ISO standard based on REA.

Weber (1986) originally empirically validated the original core REA semantic model. Investigating whether software practitioners had both identified and solved the same problems that academicians had identified, he found that the REA model fulfilled its objective as a generalized model and that it predicted the high-level semantics found in all twelve software packages he reviewed. He also suggested adding some constructs, which other scholars have since incorporated into the expanded REA ontology. Weber's research is a good example of how evaluative research provides input into design science research, which then updates the previously developed model to incorporate previously found lacking factors.

Andros, Cherrington, and Denna (1992) and Cherrington, Denna, and Andros (1996) used the REA model as a basis for system design. They found that IBM Corporation obtained significant benefits from a semantically modeled employee-reimbursement system based on the REA model. The authors reported that the new system significantly reduced IBM's accounts payable department's time to process employee reimbursements, reduced costs, and increased employee satisfaction. These studies demonstrate how one can use the REA model as the basis for system design and show how end users perceive the resultant systems, which completes the research loop from design to end user.

To better understand if businesses were adhering to REA concepts and whether REA-like systems had any advantages compared to traditional general ledger systems, David (1995) developed a metric to classify organizations' accounting systems characteristics (ASC) along a continuum between traditional general ledger-based accounting systems and REA systems. She based this metric on characteristics that she identified in theoretical research as critical characteristics for REA systems. She visited pulp and paper industries and conducted structured interviews with management to collect data on accounting systems, productivity, efficiency, and perceived competitive advantage. She then categorized the firms using the ASC metric. Consistent with Andros et al. (1992) and Cherrington et al. (1996), David (1995) found that firms that scored "more REA-like" on the ASC were associated with higher productivity and administrative efficiencies.

Haugen and McCarthy (2000) argue that REA is the appropriate model for creating a semantic Web for Internet supply chain collaboration. They state that the REA supply chain (including all resources, events, agents, and relationships among them) connects across enterprises at a higher semantic level than alternative supply chain models (e.g., ERP, EDI, APS, XML). Haugen and McCarthy explain that the REA supply chain can be implemented via XML or EAI scripting to connect ERP systems across supply chains. They identify several advanced planning and scheduling (APS) systems that are either compatible with REA or similar to REA. They also present an implemented semantic REA supply chain. Haugen and McCarthy assert that their implementation is different from most other workflow systems that simply route documents from one party to the next until the documents get completed. The REA system automatically propagates demand to the next agents in the supply chain, which business events drive. This system is among the first documented uses of REA as a semantic supply chain.

O'Leary (2004) compared REA with SAP, the leading enterprise software package at the time, and determined that SAP is consistent with the REA ontology in its database, semantic, and structuring orientations. However, he notes that SAP also contains implementation compromises (in other words, implementations that have been altered from the normative prescriptions of REA) in its structuring and semantic orientations based in part on accounting artifacts. Some of the compromises that O'Leary encountered likely result from the fact that software vendors' products evolve in an incremental fashion and that vendors rarely design or redesign software starting with a clean slate. Because SAP and similar software vendors originally created their accounting modules based on the general ledger and its related artifacts, their incremental modifications were unlikely to alter the underlying foundation but would instead change surface features. The general ledger still forms the basic structure of SAP's financial accounting module. By evolving slowly instead of re-engineering their software, vendors such as SAP provide consistency from version to version and protect their installed user base.

Consistent with O'Leary (2004), Fallon and Polovina (2013) found that the human capital management (HCM) module in SAP conforms to the REA ontology. They also reported that one can use REA for modeling the HCM business processes in SAP. Interestingly and consistent with McCarthy (1982), they also identified potential problems resulting from SAP's deviation from REA's ontology. These problems relate to inconsistency of data, information gaps, and overlaps of data resulting from how SAP stores information in its database. For example, Fallon and Polovina report that SAP does not store hire event data in a single table. Rather, SAP initially stores the data in an applicant table, and, if the applicant is hired, SAP moves the data to the employee table. As a result, one loses the ability to trace the event at the point of "hiring". Fallon and Polovina attribute this implementation compromise to the cost of storage and processing speed. They also identify other areas in which SAP does not fully conform to REA as Dunn (2012) presents, such as direct relationships between agents and resources (instead of linking the agents and resources to an event in which they both participate) and events lacking responsible agents. Fallon and Polovina conclude that SAP, while basically consistent with REA, could benefit from an even higher level of consistency to reduce data redundancy and loss. Similarly, they state that REA could benefit from including additional event entities such as are found in existing ERP systems that would allow the ontology to more fully represent the HCM environment. The additional events they advocate including are in fact part of REA's workflow/task level.

Likewise, in the HCM domain, Sutheparaks, Vatanawood, and Patanothai (2011) use REA to develop a global schema for the extract, transform, and load (ETL) methodology (which is part of a business intelligence architecture). Using REA in this manner creates a conceptual basis for identifying the needed data and its appropriate organization so that one can then implement it into business intelligence systems. They claim that the REA's embedded semantics and business patterns allow developers (who do not necessarily have domain expertise) to obtain the appropriate information and develop more accurate requirement specifications. They performed a case study in a university setting to demonstrate a proof of concept. They found that the two levels of structure provided by REA (policy and operational) were critical to allow the development of the ETL global schema. They converted the developed conceptual model to RDF using Protégé and transformed the RDF data into a normalized relational schema using the R2D (RDF-to-relational) framework. Then, they used an open source ETL solution to implement the ETL activities (including cleansing, normalizing, etc.). The researchers claim that using the REA ontology provides increased understanding and accuracy in the ETP process and the development of a global schema for the HCM area.

Curry (2009) also reports on the benefits of an REA implementation of HR and financial modules at an enterprise comprising four separate company types (medical staffing, business brokerage, property management, and IT consulting). The enterprise built the system around objects and not accounting artifacts. It took only 63 days to implement both the HR and financial modules and less than two days to add a completely new company. One benefit management observed was a significant savings in the total cost of ownership (US$400,000 in savings as of the report date and an additional US$200,000 in expected savings). Another benefit was that users (including those who initially resisted the change to a new system) found the new system to be intuitive and supportive and found that it provided more meaningful information at the appropriate level of detail than did their previous general ledger-based system.

While researchers have reported savings and increased ease of use with REA (e.g., Curry, 2009) Vandenbossche and Wortmann (2006) examine why organizations do not more fully incorporate REA into ERP systems. They state that ERP systems view DCA as the basic accounting model that provides data for other applications, and, therefore, ERP systems include general ledger modules to accomplish accounting tasks. Vandenbossche and Wortmann (2006) rightly observe that most accountants and other financial users want ERP systems to follow the DCA approach because that is what they are accustomed to using. Although O'Leary (2004) and Fallon and Polovina (2013) found SAP to be consistent with the REA ontology's database, semantic, and structuring orientations, despite some implementation compromises the authors observed, Vandenbossche and Wortmann (2006) state that REA has conceptually solved complexities in current ERP data models, but its use would require a new ERP data model. Vandenbossche and Wortman's claim is consistent with that of Grabski, Leech, and Schmidt (2011) who view the REA ontology as a revolutionary rather than evolutionary approach in the development of ERP systems. Converting to an REA-ontology based system requires a completely fresh start. Workday (www.workday.com) had a completely fresh start. Whereas most other ERP software was originally created by building connections between existing software packages—in essence gluing together accounting, materials resource planning, human resources, and other software packages—Workday started with no existing software. Workday was not tied to the old way of structuring systems; rather, it developed its "beyond ERP" software on REA to provide support for financial, resource, and revenue management. Workday's developers realized the potential for enterprise software founded on an ontologically modeled database, and used REA as the foundation of Workday's software. They claim to address the requirements of accounting, risk management, corporate governance, and analytics in a single cohesive integrated system (Workday, 2010).

The international standard ISO/IEC 15944-4 (ISO/IEC, 2007) builds on the REA design theory and describes both the independent view of inter-enterprise events and also the trading partner views (upstream and downstream) of the inter-enterprise events. The standard also incorporates various business states such as waiting start, in-service, completed, aborted, materialized, planned, specified, pending, proposed, and so forth. The ISO/IEC 15944-4:2007 standard provides the ontological specification needed in an economic exchange, and it is the declarative component of the Open-edi Business Transaction Ontology (OeBTO). Figure 6 shows the Open-edi Business Transaction Ontology6. Standards work with the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) has also used the REA ontology work.

Zdravkovic and Ilayperuma (2010) examine the relationship between REA and the Open-edi business framework for service modeling and propose a service-centric business model. They use the Open-edi Business Transaction Ontology (OeBTO), developed by the ISO, to create a service-oriented architecture model for e-services. They extend the OeBTO model for a service-related issues and include the value-chain and business events (see Figure 7). They say the major strength of their proposed method is that it combines the multiple layers of REA and OeBTO to identify an entire enterprise-wide service portfolio on the business level that is well-enough defined to be transformed further to a system-centric e-service model.

Laurier and Poels (2012a) used the ISO standard 15944-4 (ISO/IEC, 2007) to illustrate how one can track and trace product and monetary flows. Specifically, they show how one can track both intra- and inter-organizational enterprise phenomena in a prototype application of a pizza bakery's supply chain from the farm to the customer to demonstrate the robustness of the REA ontology. Such an agreed-on ontology-based standard is crucial for facilitating inter-organizational system development. Hunka, Zacek, Melis, and Sevcik (2011) also use REA to model a supply chain and to integrate the systems of trading partners in it.

REA allows one to define and use abstracted business processes as a set of patterns for designing business applications (Hruby 2006; Laurier & Poels 2012b); as such, it promises to be flexible enough to adapt to specific enterprise needs while providing a solid foundation for improving software quality. Researchers have found the REA ontology to support many, but not all, of the balanced scorecard and strategic enterprise management (SEM) constructs (Grabski, Leech, & Schmidt, 2011), which is consistent with Curry's (2009) observation that REA ontology's focus on business processes prevalent throughout all enterprises enables them to shape it to their needs.

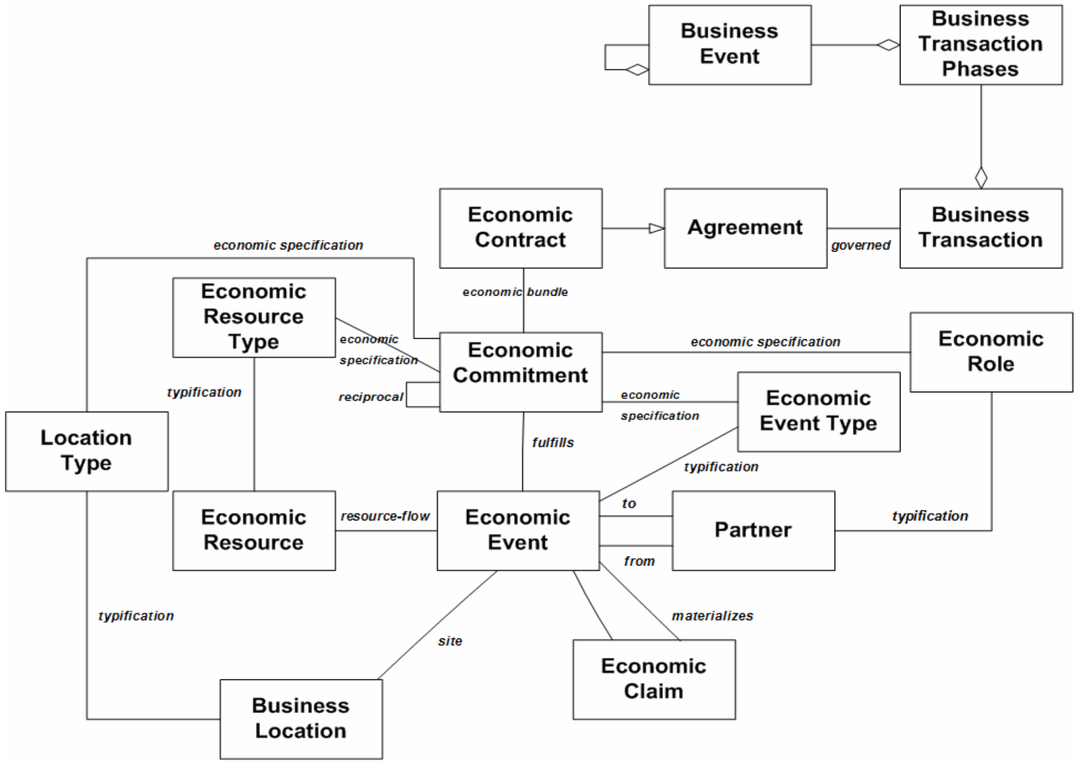

Figure 5. Open EDI Ontology with Business Transaction Phases and Business Events (ISO/IEC, 2007, p. 27)

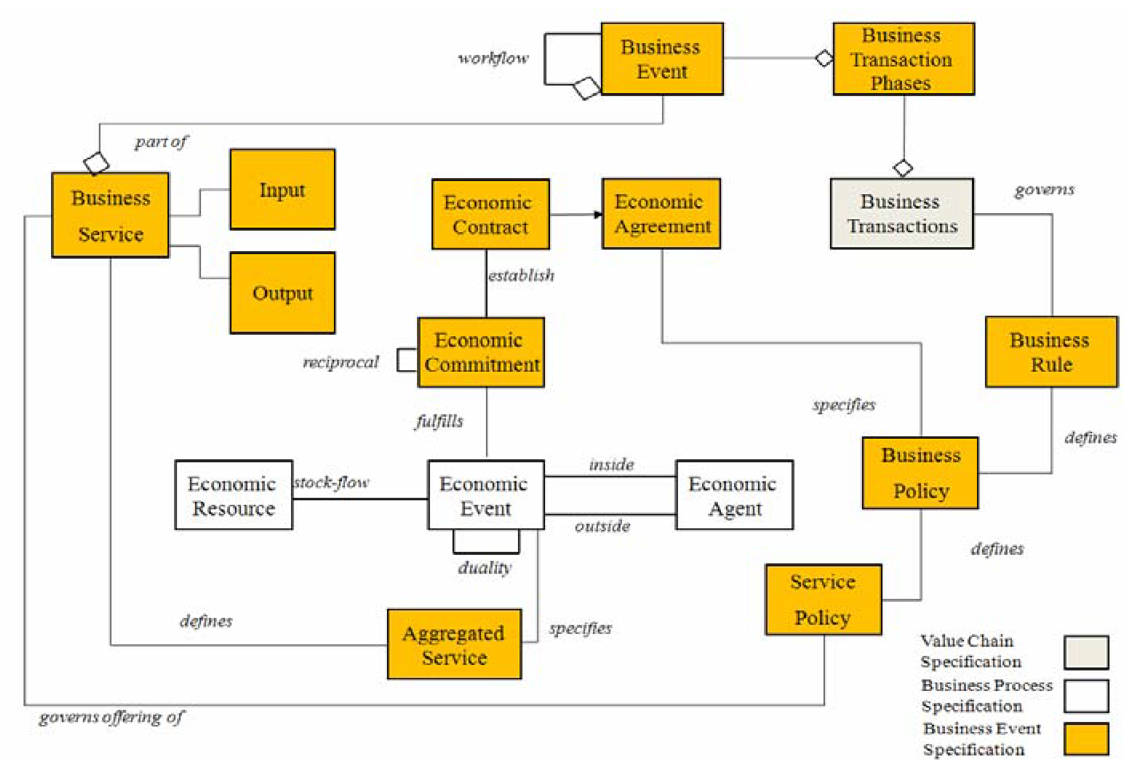

Figure 6. Service-oriented Architecture Model for E-services (Zdravkovic & Ilayperuma, 2010, p. 114)

6.1 Summary of the Influence of REA on Practice

REA has a significant impact on practice. Research has found enterprise systems are consistent with REA and have also identified shortcomings in current implementations they could have avoided if they had implemented the full REA ontology. Perhaps this is why we now see an ISO standard that incorporates the REA ontology for inter-organizational trading partners. One can overcome the aforementioned limitations by adopting this standard. Finally, REA serves as the basis for a “new” enterprise system—one that was built from the ground up rather than needing to keep compatibility with an installed user base. Again, one can overcome the limitations identified in the research in this type of implementation.

7 REA Going Forward and Concluding Comments

What might the future hold for the REA design theory? We encourage advancement along two fronts. First, we encourage researchers to use new technological advances and other research areas to further develop and expand REA’s capabilities. Second, we encourage researchers to use REA design theory to enhance and further develop other research areas. We anticipate systems in practice will increasingly incorporate foundational constructs that are consistent with REA design theory. However, researchers must monitor the systems and technologies in practice to determine whether new enhancements and developments have implications for REA research. Based on research already conducted, we believe researchers will extend REA design theory to enable it to represent an independent view of multiple enterprises, to demonstrate the ability to integrate REA-based systems with unstructured data (both from within and outside the enterprise system), to transform corporate financial reporting to a form more consistent with REA constructs, and to also contribute to transforming assurance services from a practice that involve tests of small samples of transactions to a practice that incorporates entire datasets. We also believe researchers can use REA to help their research in other areas. For example, perhaps the previously discussed e^3value research (Buder & Felden, 2012) has benefited from considering existing REA research and vice versa. Perhaps incorporating REA into research with the semantic technology/upper ontology language OWL (Gailly & Poels, 2007a; Sedbrook & Newmark, 2010; Sedbrook, 2010) has enhanced OWL. Practitioners will continue to adopt REA constructs and demonstrate them as robust or identify needs for improvement. REA constructs are an integral part of international commerce standard ISO/IEC 15944-4 (2007). They have been incorporated into enterprise systems such as Workday, which has published many case studies in its corporate blogs demonstrating unprecedented system agility (see, e.g., Swete, 2012, 2013).

Changes in the way enterprises do business necessitate system agility because the system must be able to evolve with the business such as the example in Curry (2009) (see Section 6). Systems based on the semantics of an enterprise’s underlying reality are more agile than are systems founded on artificial constructs (McCarthy, David, & Sommer, 2003). Today’s business climate is filled with change. Enterprises are increasingly involved in collaborations that defy traditional corporate boundaries. The ability to innovate or imagine new business models is one of the most valued traits an executive can have. Partnering organizations that build their enterprise databases on the REA ontology will be better able to integrate their databases, which they will probably best accomplish by using object technology and artificial intelligence concepts such as automated intensional reasoning (Geerts & McCarthy, 2000a) and automated intelligent agents. Automated intensional reasoning systems make inferences based on database table intensions and require system designers to consistently adhere to any underlying pattern or design theory such as REA. For this to work well, future REA research will need to address previously mentioned concerns about compromise while finding a way to remain agile.

Recent technological advances increase the importance of founding enterprise systems on a design theory such as REA that uses constructs that mirror the underlying business objects rather than the general ledger. Increased processing speed and solid state drives have become available at low enough cost that databases can now be stored and accessed in memory. Data warehouses will likely become obsolete because queries may be run against the operational database without hampering the system’s ability to capture and store additional transactions. Sisco (2015) describes its software’s capability for such querying. Similarly, management accountants will soon demand access to raw transaction data rather than having to derive and estimate from general ledger accounts the numbers they need to support management decisions. As Swete (2012, 2013) allude, the metadata in an agile system should mirror real-world business objects, whereas the specific database technology used should change continuously to keep up to date with contemporary developments.

Software as a service, enabled by cloud computing, has been quickly gaining momentum in the business world. Gartner (2013) says that cloud computing is an inevitable trend and that companies must develop cloud strategies to successfully compete in the future. Gartner also says software providers should take steps toward agile, loosely coupled, more self-sufficient systems with backend integration to other corporate business processes. Social media is also prevalent in the business world. Companies are practically drowning in the huge volumes of data they are gathering from social media networks. This situation highlights enterprises' increased need for integrating unstructured data sources with structured databases because social media is primarily unstructured. The more closely the structured database foundational constructs resemble the underlying reality, the easier it will be to integrate unstructured data. O'Leary (2015) identifies REA as the foundation for an architecture to integrate cloud computing, accounting, and enterprise systems using the public/private processes of RosettaNet as the conceptual basis to capture information used in the cloud. He uses Workday as a case study with its customer-defined worktags that are tagged to transactional data and are used to identify key dimensions of the business that management would like to track and analyze, such as customer, product, region, and project. We expect additional research to elaborate on tagging transactional data using REA construct tags to enable various aggregations for different types of decisions.

Dunn (2004) speculates that organizations could use the REA ontology to transform the practice of corporate financial reporting and suggests the current balance sheet, income statement, and statement of cash flows should be supplemented with a statement of resources and resource flows, a statement of economic and commitment events, and a statement of agents. Similarly, Citak and Gal (2010) claim the ISO 15944-4 framework allows for more diverse reporting opportunities than do current financial statements because frameworks that embody more semantics are better than those that offer less semantics. Other researchers suggest using REA to extend the XBRL GL taxonomy to provide uniform access to information and more reporting and query permutations to facilitate better and more timely business reporting (Amrhein, Farewell, & Pinsker, 2009; Amrhein, 2011). We believe that, for the combination of XBRL and REA to be successful, tagging must not be done at the reporting level but instead applied to transactions, similar to the Worktags that Workday uses (Nittler, 2012; O'Leary, 2015).

Another issue is that of preserving the semantics at an operational level beyond the level of the database itself to allow decision makers additional insight into the problems and the information available to address the issues that they face. Limited research has examined the similarities of the semantic models underlying current ERP packages. Nonetheless, these models do exist, and many organizations reengineer themselves to become consistent with the best practices embodied in these models. Unfortunately, organizations do so often at the workflow level and lose the benefits of the underlying semantics. This lack of semantics is apparent when organizations seek to extend their value chains up and down their supply chain. Preserving the underlying semantics and standardizing semantic patterns enable automated intensional reasoning and other knowledge-based tools to facilitate inter-enterprise trade. Semantically modeled enterprise information systems will provide many benefits from the individual decision maker level to the inter-organizational level. The critical issue is to ensure that the semantics are not lost on implementing the system and obscured by the task-level mechanics. When this occurs, all subsequent benefits are lost, and we are faced with the task of integrating disparate systems that are conceptually identical. We hope to see much thoughtful research on how ontologically modeled databases may help transform corporate reporting because the number of surprise bankruptcies and financial scandals in the past decades evidences that the current reporting model is broken.

Financial accounting is not the only area in which we expect to see applications of REA increase and transform reporting. Church and Smith (2007) found that the REA ontology supports most of the balanced scorecard information requirements and suggest extending REA to include the remaining balanced scorecard information requirements and nonfinancial measures needed for other management systems. Church and Smith (2008) propose using REA ontology-based simulation models to facilitate strategic planning. The REA-based dynamic models provide the basic patterns needed to support a variety of management planning tasks. Clinton and Van Der Merwe (2008) indicate that REA can provide the details needed for a management accounting approach called resource consumption accounting (RCA), with REA providing natural support for RCA's value chain and value layers (for external reporting and for decision support).

Assurance is another area of accounting that may benefit from REA in the future. Gal, Geerts, and McCarthy (2009) illustrate how one may semantically specify and automatically enforce internal control procedures in REA-based accounting systems. Weigand and Elsas (2012) formalize an audit approach that one can use with REA models. Dunn (2012) illustrates how one can use the REA ontology as a framework for assessing risk when evaluating a company's internal controls. Perols and Murthy (2012) present an information-fusion architecture with specific components based on REA, machine-learning and the continuous assurance literature.

REA has been especially useful in supply chain research (e.g., Haugen & McCarth, 2000; Hunka et al., 2011), and we expect to see REA research continue in this area. Geerts and O'Leary (2014) recently created the EAGLET ontology based on REA constructs to demonstrate the traceability of individual objects in a "supply chain of things". Geerts and O'Leary's research demonstrates REA's capability to integrate internal transaction data with external data such as that generated by machines and appliances in the Internet of things, a promising future research opportunity.

As individuals imagine new business models and invent new technologies, the world needs enterprise systems that are agile enough to adapt and that have a high degree of interoperability. Business model innovations are tending toward more collaboration and federation that inter-enterprise systems will facilitate. Databases modeled according to a design theory that provides a common vocabulary for all users should serve as the foundation for integrated enterprise and inter-enterprise systems. The REA design theory is a robust candidate for said databases, especially when combined with semantic technology. Enterprises can facilitate system integration in and between themselves by using common semantic patterns that intelligent systems can reason about. This use of common semantic patterns can result in business partner companies integrating their systems without using identical workflow business practices.

Originally developed as a generalized accounting framework, REA has evolved into a robust design theory that encompasses all business processes and is relevant to all enterprise systems researchers. Extensions to REA since its origin include type images; enterprise value chains and workflow/task specification; and policy-level, commitment-level, and operational-level information integration. The REA design theory is a theoretically based domain ontology for reporting enterprises' economic story and is the basis for the ISO/IEC standard on economic exchanges between organizations. However, REA's impact goes beyond theory: the REA ontology is also the foundation for a cloud-based enterprise system (www.workday.com) that has thus far enabled adopters to be more nimble than with their old systems. Researchers have even found legacy ERP systems (e.g., SAP) to be consistent with REA; however, implementation compromises made in vendor ERP software have resulted in some data inconsistencies, overlaps of data, and information gaps. Researchers have found that REA models improve communication between analysts and users and that using REA conceptual models results in a higher perceived ease of use than does using other conceptual models. Going forward, researchers need to be aware of REA's benefits and limitations to contribute to its advancement by engaging in future REA research. In this paper, we identify several contemporary developments in technology that represent opportunities for REA design science research. However, we hope readers will consider other opportunities based on their own research agendas. Such opportunities may result from identifying means by which REA constructs may advance readers' other research, or they may result from identifying means by which readers' other research may advance REA design theory.

References

Abrial, J. R. (1974). Data semantics. In J. W. Klimbie & K. L. Koffeman (Eds.), *Data base management* (pp. 1-60). Amsterdam: North Holland.

Allen, G. N., & March, S. T. (2003). Modeling temporal dynamics for business systems. *Journal of Database Management, 14*(3), 21-36.

Allen, G. N., & March, S. T. (2006). *A critical assessment of the Bunge-Wand-Weber ontology for conceptual modeling.* Paper presented at the Workshop on Information Technologies and Systems, Milwaukee, WI, USA.

Amrhein, D. G. (2011). Integrating REA and XBRL GL to facilitate modern business reporting. *International Journal of Business and Social Science, 2*(24), 1-11.

Amrhein, D. G., Farewell, S., & Pinsker, R. (2009). REA and XBRL GL: Synergies for the 21st century business reporting system. *The International Journal of Digital Accounting Research, 9*, 127-152.

Andros, D., Cherrington, J. O., & Denna, E. L. (1992). Reengineer your accounting the IBM way. *The Financial Executive,* 8(4), 28-31.

Ariely, D. (2008). *Predictably irrational: The hidden forces that shape our decisions* (1st ed.). New York, NY: Harper.

Armitage, H. M. 1985. *Linking management accounting systems with computer technology*. Hamilton, Ontario: Society of Management Accountants of Canada

Bachman, C. W. (1965). Integrated data store. *DPMA Quarterly*.

Batra, D., & Sin, T. (2008). The READY model: Patterns of dynamic behavior in REA-based accounting applications. *Information Systems Management,* 25(3), 200-210.

Black, J. D., & Black, A. G. (1929). *Production organization*. New York: Henry Holt and Company.

Bodart, F., Patel, A., Sim, M., & Weber, R. (2001). Should optional properties be used in conceptual modeling? A theory and three empirical tests. *Information Systems Research, 12*(4), 384-405.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modeling Language reference manual*, Reading, MA: Addison-Wesley.

Borch, S. E., Stefansen, C. (2004). Evaluating the REA enterprise ontology from an operational perspective. In *Proceedings of the Caise'04 Workshops*.

Bowen, P. L., O'Farrell, R. A., & Rohde, F. H. (2006). Analysis of competing data structures: Does ontological clarity produce better end user query performance. *Journal of the Association for Information Systems, 7*(8), 514-544.

Bubenko Jr., J.A. (1976). *The temporal dimension in information modelling* (research report RC-6187). Yorktown Heights, NY: IBM Research Laboratories.

Buder, J., & Felden, C. (2012). Evaluating business models: Evidence on user understanding and impact to BPM correspondence. In *Proceedings of the 45th Hawaii International Conference on System Sciences* (pp. 4336-4345).

Chamberlin, D. D., Astrahan, M. M., Eswaran, K. P., Griffiths, P. P., Lorie, R. A., Mehl, J. W., Reisner, P., & Wade, B. W. (1976). SEQUEL 2: A unified approach to data definition, manipulation, and control. *IBM Journal of Research and Development*, *20*(6), 560-575.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology*, *5*, 55-81.

Chen, P. P. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems*, *1*(1), 9-36.

Chen, J. L., McLeod, D., & O'Leary, D. (1995). Domain-knowledge-guided schema evolution for accounting database systems. *Expert Systems with Applications*, *9*(4), 491-501.

Chen, J. L., McLeod, D., & O'Leary, D. (1998). REA accounting database evolution. In M. A. Vasarhelyi & A. Kogan (Eds.) *Artificial intelligence in accounting and auditing* (vol. 4, pp. 139-159). New York: Marcus Wiener.

Cherrington, J. O., Denna, E. L., & Andros, D. P. (1996). Developing an event-based system: The case of IBM's national employee disbursement system. *Journal of Information Systems*, *10*(1), 51-69.

Church, K., & Smith, R. (2007). An extension of the REA framework to support balanced scorecard information requirements. *Journal of Information Systems*, *21*(1), 1-25.

Church, K., & Smith, R. (2008). REA ontology-based simulation models for enterprise strategic planning. *Journal of Information Systems, 22*(2), 301-329.

Citak, N., & Gal, G. (2010). Will better information be a solution to better management and more transparency for investors? A look at ISO 15944. *Modav, 1*, 183-208.

Clinton, B. D., & Van Der Merwe, A. (2008). Understanding resource consumption and cost behavior—part II. *Cost Management, 22*(4), 14-20.

Coad, P. (1995). *Object models: Strategies, patterns, & applications*. Englewood Cliffs, NJ: Prentice-Hall.

CODASYL. (1971). *Data base task group report*. New York: ACM.

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, *13*(6), 377-387.

Colantoni, C. S, Manes, R. P., & Whinston, A. (1971). A unified approach to the theory of accounting and information systems. *The Accounting Review, 46*(1), 90-102.

Correia, J. (2013). Software & cloud business models: Convergence or disruption? *Gartner.*

Curry, S. C. (2009). *REA: A process-oriented approach to enterprise systems* (Unpublished Master Thesis). Michigan State University.

David, J. S. (1995). *An empirical analysis of REA accounting systems, productivity, and perceptions of competitive advantage* (Unpublished Doctoral Dissertation). Michigan State University.

David, J. S. (1997). Three "events" that define a REA methodology for systems analysis, design and implementation. In *Proceedings of the Annual Meeting of the American Accounting Association*.

David, J. S., Dunn, C. L., & McCarthy, W. E. (1999). Enterprise resource planning systems research: The Necessity of explicating and examining patterns in symbolic form. In *Proceedings of the 1st International Workshop on Enterprise Management and Resource Planning Systems*.

David, J. S., Gerard, G. J., & McCarthy, W. E. (2002). Design science: An REA perspective on the future of AIS. In V. Arnold & V. G. Sutton (Eds.), *Researching accounting as an information systems discipline* (pp. 35-63). Sarasota, FL: American Accounting Association.

David, J. S., Dunn, C. L., McCarthy, W. E., & Poston, R. S. (1999). The research pyramid: A framework for accounting information systems research. *Journal of Information Systems*, *13*(1), 7-30.

Denna, E. L., Cherrington, J. O., Andros, D. P., & Hollander, A. S. (1993). *Event driven business solutions*. Burr Ridge, IL: Irwin Professional Publishing.

Denna, E. L., Jasperson, J., Fong, K., & Middleman, D. (1994). Modeling conversion process events. *Journal of Information Systems*, *8*(1), 43-54.

Denna, E. L., & McCarthy, W. E. (1987). An events accounting foundation for DSS implementation. In C. W. Hosapple & A. B. Whinston (Eds.), *Decision support systems: Theory and application* (pp. 239-263). Berlin, Germany: Springer-Verlag.

Dey, D., Barron, T., & Storey, V. (1995). A conceptual model for the logical design of temporal databases. *Decision Support Systems, 15*(4), 305-321.

Dunn, C. L. (1994). *An abstraction hierarchy as a database interface: Does it control complexity?* (Unpublished doctoral dissertation). Michigan State University.

Dunn, C. L. (2004). *Transforming the practice of corporate financial reporting*. Paper presented at the First REA Technology Workshop, Copenhagen, Denmark.

Dunn, C. L. (2012). *REA accounting systems: Resources-events-agents: An ontology for designing, controlling, and using integrated enterprise systems*. Illinois: McGraw-Hill Create.

Dunn, C. L. (2013). *Why teach resources-events-agents (REA) accounting? After all, no one uses it!* Paper presented at the AIS Educators Conference, Laramie, Wyoming.

Dunn, C. L., & Gerard, G. J. (2001). Auditor efficiency and effectiveness with diagrammatic and linguistic conceptual model representations. *International Journal of Accounting Information Systems*, *2*(4), 223-248.

Dunn, C. L., & Grabski, S. V. (1998). The effect of field independence on conceptual modeling performance. *Advances in Accounting Information Systems*, *6*, 65-77.

Dunn, C. L., & Grabski, S. V. (2000). Perceived semantic expressiveness of accounting systems and task accuracy effects. *International Journal of Accounting Information Systems*, *1*(2), 79-87.

Dunn, C. L., & Grabski, S. V. (2001). An investigation of localization as an element of cognitive fit in accounting model representations. *Decision Sciences, 32*(1), 55-94.

Dunn, C. L., & Grabski, S. V. (2002). Empirical research in semantically modeled accounting systems. In V. Arnold & S. G. Sutton (Eds.), *Researching accounting as an information systems discipline* (pp. 157-180). Sarasota, FL: American Accounting Association.

Dunn, C. L., & McCarthy, W. E. (1997). The REA accounting model: intellectual heritage and prospects for progress. *Journal of Information Systems*, *11*(1), 31- 51.

Dunn, C. L., Gerard, G. J., & Grabski, S. V. (2011). Diagrammatic attention management and the effect of conceptual model structure on cardinality validation. *Journal of the Association for Information Systems*, *12*(8), 585-605.

Dunn, C. L., Gerard, G. J., & Grabski, S. V. (2013). *Cognitive fit and user schemas: An expanded model for problem-solving.* Paper presented at the Pre-ICIS Workshop on Accounting Information Systems, Orlando, Florida.

Dunn, C. L., Gerard, G. J., Grabski, S. V., & Boss, S. R. (2013). *Asymmetry in identification of multiplicity errors.* Paper presented at the Pre-ICIS Workshop on Accounting Information Systems, Milan, Italy.

Everest, G. C., & Weber, R. (1977). A relational approach to accounting models. *The Accounting Review*, *52*(2), 340-359.

Fallon, R., & Polovina, S. (2013). REA analysis of SAP HCM; some initial findings. In *Proceedings of the 3rd Cubist Workshop* (pp. 31-43).

Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). Methontology: From ontological art towards ontological engineering. In *Ontological Engineering AAAI-97 Spring Symposium Series*. Stanford: AAAI Press.

Fisher, I. (1906). *The nature of capital and income*. New York: MacMillan.

Fowler, M. (1997). *Analysis patterns: Reusable object models*. Reading, MA: Addison-Wesley.

Gailly, F., & Geerts, G. L. (2013). Ontology-driven business rule specification. *Journal of Information Systems*, *27*(1), 79-104.

Gailly, F., & Poels, G. (2007a). Ontology-driven information systems: Redesign and formalization of the REA ontology. In *Lecture Notes in Computer Science* 4439 (pp. 245-259).

Gailly, F., & Poels, G. (2007b). Ontology-driven business modeling: Improving the conceptual representation of the REA ontology. In *Conceptual Modeling—ER 2007 Lecture Notes in Computer Science 4801* (pp. 407-422).

Gailly, F., Laurier, W., & Poels, G. (2008). Positioning and formalizing the REA enterprise ontology. *Journal of Information Systems, 22*(2), 219-248.

Gal, G., Geerts, G. L., & McCarthy, W. E. (2009). *Semantic specification and automated enforcement of internal control procedures within accounting systems*. Paper presented at the 3rd Value Modeling and Business Ontologies Workshop, Stockholm.

Gal, G., & McCarthy, W. E. (1983). Declarative and procedural features of a CODASYL accounting system. In P. Chen (Ed.) *Entity-relationship approach to information modeling and analysis* (pp. 197-213). Amsterdam, The Netherlands: North-Holland.

Gal, G., & McCarthy, W. E. (1985). Specification of internal controls in a database environment. *Computers and Security*, *4*(1), 23-32.

Gal, G., & McCarthy, W. E. (1986). Operation of a relational accounting system. *Advances in Accounting*, *3*, 83-112.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. Reading, MI: Addison Wesley.

Geerts, G. L. (2008). Introduction to the REA 25th anniversary special section. *Journal of Information Systems, 22*(2), 215-217.

Geerts, G. L., Graham, L. E., Mauldin, E. G., McCarthy, W. E., & Richardson, V. J. (2013). Integrating information technology into accounting research and practice. *Accounting Horizons*, *27*(4), 815-840.

Geerts, G. L., & McCarthy, W. E. (1997a). Modeling business enterprises as value-added process hierarchies with resource-event-agent object templates. In J. Sutherland & D. Patelp (Eds.), *Business object design and implementation* (pp. 94-113). Berlin, Germany: Springer-Verlag.

Geerts, G. L., & McCarthy, W. E. (1997b). *Accounting as romance: Patterns of unrequited love and incomplete exchanges in life and in business software*. Paper presented at the ACM Conference on Object-Oriented Programming Systems, Languages, and Applications, Atlanta, GA.

Geerts, G. L., & McCarthy, W. E. (1999). An accounting object infrastructure for knowledge-based enterprise models. *IEEE Intelligent Systems and Their Applications, 14*(4), 89-94.

Geerts, G. L., & McCarthy, W. E. (2000a). Augmented intensional reasoning in knowledge-based accounting systems. *Journal of Information Systems*, 1*4(*2), 127-150.

Geerts, G. L., & McCarthy, W. E. (2000b). The ontological foundation of REA enterprise information systems. In *Proceedings of the Annual Meeting of the American Accounting Association* (pp. 127-150).

Geerts, G. L., & McCarthy, W. E. (2002). An ontological analysis of the primitives of the extended-REA enterprise information architecture. *International Journal of Accounting Information Systems*, *3*(1), 1-16*.*

Geerts, G. L., & McCarthy, W. E. (2006). Policy-level specifications in REA enterprise information systems. *Journal of Information Systems, 20*(2), 37-63.

Geerts, G. L., McCarthy, W. E., & Rockwell, S. R. (1996). Automated integration of enterprise accounting models throughout the systems development life cycle. *International Journal of Intelligent Systems in Accounting, Finance and Management*, *5*(3), 113-128.

Geerts, G. L., & O'Leary, D. E. (2014). A supply chain of things: The EAGLET ontology for highly visible supply chains. *Decision Support Systems*, *63*, 3-22.

Gemino, A., & Wand, Y. (2005). Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties. *Data & Knowledge Engineering, 55*(3), 301-326.

Gerard, G. J. (2005). The REA pattern, knowledge structures, and conceptual modeling performance. *Journal of Information Systems*, *19*(2), 57-77.

Goetz, B. E. (1939). What's wrong with accounting? *Advanced Management, 4*(5), 151-157.

Goetz, B. E. (1949). *Management planning and control*. New York, NY: McGraw-Hill.

Grabski, S. V., Leech, S. A., & Schmidt, P. J. (2011). A review of ERP research: A future agenda for accounting information systems. *Journal of Information Systems, 25*(1), 37-78.

Grabski, S. V., & Marsh, R. J. (1994). Integrating accounting and manufacturing information systems: An ABC and REA-based approach. *Journal of Information Systems*, *8*(2), 61- 80.

Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association for Information Systems*, *8*(2), 312-225.

Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, *5*(2), 199-220.

Guan, J., Cobb, A., & Levitan, A. (2006). An ontological model for accounting information systems. In *Proceedings of the Americas Conference on Information Systems*.

Guarino, N. (1997). Understanding, building and using ontologies. *International Journal of Human-Computer Studies, 46,* 293-310.

Hammer, M. (1990). Reengineering work: Don't automate, obliterate. *Harvard Business Review, 68*(4), 104-112.

Hammer, M., & Champy, J. (1993). *Reengineering the corporation*. New York: Harper Collins.

Haseman, W. D., & Whinston, A. B. (1976). Design of a multidimensional accounting system. *The Accounting Review, 51*(1), 65-79.

Haseman, W. D., & Whinston, A. B. (1977). *Introduction to data management*. Rockport, Il: R.D. Irwin.

Haugen, R., & McCarthy, W. E. (2000). *REA: A semantic model for internet supply chain collaboration*. Paper presented at the The ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, Minneapolis, Minnesota, USA.

Hay, D. (1996). *Data model patterns*. New York, NY: Dorset House Publishing.

Hessellund, A. (2006). *Modeling issues in REA*. Paper presented at the 2nd International REA Technology Workshop, Santorini Island, Greece.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-105.

Holsapple, C. W., & Whinston, A. B. (1984). Building blocks for decision support systems. In *Proceedings of the NYU Symposium: New Directions for Database Systems* (pp. 121-164).

Hruby, P. (2003). Universal enterprise model: Business pattern language. In *Proceedings of the First Nordic Conference on Pattern Languages of Programs.*

Hruby, P. (2005). *Role of domain ontologies in software factories*. Paper presented at the International Workshop on Software Factories at OOPSLA 2005, San Diego, California, USA.

Hruby, P. (2006). *Model-driven design using business patterns*. New York; Berlin: Springer-Verlag.

Hruby, P., & Kiehn, J. (2006). *Revised classification of the enterprise information architecture elements*. Paper presented at the 3rd International Conference on Enterprise Systems and Accounting, Santorini Island, Greece.

Hunka, F., & Zacek, J. (2014). Detailed analysis of REA ontology. *Advances in Enterprise Engineering VIII* (pp. 61-75). New York: Springer.

Hunka, F., Zacek, J., Melis, Z., & Sevcik, J. (2011). REA value chain and supply chain. *Scientific Papers of the University of Pardubice. Series D, Faculty of Economics and Administration*.

Ijiri, Y. (1967). *The foundations of accounting measurement: A mathematical, economic, and behavioral inquiry*. Englewood Cliffs, NJ: Prentice Hall.

Ijiri, Y. (1975). *Theory of accounting measurement*. Sarasota, Florida: American Accounting Association.

ISO/IEC. (2007). *ISO 15944-4: Information technology—business operational view—part 4: Business transaction scenarios—accounting and economic ontology*. Geneva.

Jacobson, I. (1992). *Object-oriented software engineering: A use case driven approach*. Addison-Wesley: Reading, MA.

Jacobson, I., Ericsson, M., & Jacobson, A. (1994). *The object advantage: Business process reengineering with object technology*. New York. NY: ACM Press.

Jaquet, M. (2006). *A property driven approach towards describing semantics of REA entities.* Paper presented at the 2nd International REA Technology Workshop, Santorini Island, Greece.

Lancaster, K. J. (1975). Socially optimal product differentiation. *American Economic Review*, *65*(4), 567-585.

Laurier, W., & Poels, G. (2012a). Track and trace future, present, and past product and money flows with a resource-event-agent model. *Information Systems Management, 29*(2), 123-136.

Laurier, W., & Poels, G. (2012b). Ontology-based structuring of conceptual data modeling patterns. *Journal of Database Management, 23*(3), 50-64.

Laurier, W., & Poels, G. (2013). Invariant conditions in value system simulation models. *Decision Support Systems*, *56,* 275-287.

Lieberman, A. Z., & Whinston, A. B. (1975). A structuring of an events-accounting information system. *The Accounting Review, 50*(2), 246-258.

Lum, V., Ghosh, S., Schkolnick, M., Jefferson, D., Su, S., Fry, J., Teorey, T., & Yao, B. (1979). *1978 New Orleans data base design workshop report* (Research Report RJ2554). San Jose, CA: IBM Research Laboratories.

Maes, A., & Poels, G. (2007). Evaluating quality of conceptual modelling scripts based on user perceptions. *Data & Knowledge Engineering*, *63*(3), 701-724.

March, S. T., & Smith, G. (1995). Design and natural science research on information technology. Decision Support Systems, *15*(4), 251-266.

Mattessich, R. (1964) *Accounting and analytical methods*. Homewood, IL: Richard D. Irwin.

Mayrhofer, D. (2010). *Reference modeling for inter-organizational systems*. Paper presented at the MODELS 2010 Doctoral Symposium.

McCarthy, J., & Hayes, P. (1968). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer & D. Michie (Eds.), *Machine intelligence 4*. Edinburgh: Edinburgh University Press.

McCarthy, W. E. (1978). *A relational model for events-based accounting systems* (Doctoral Dissertation). University of Massachusetts.

McCarthy, W. E. (1979). An entity-relationship view of accounting models. *The Accounting Review*, *54*(4), 667-686.

McCarthy, W. E. (1980). Construction and use of integrated accounting systems with entity-relationship modeling. In P. Chen (Ed.) *Entity-relationship approach to systems analysis and design* (pp. 625-637). Amsterdam, The Netherlands: North Holland Publishing Company.

McCarthy, W. E. (1982). The REA accounting model: A generalized framework for accounting systems in a shared data environment. *The Accounting Review*, *57*(3), 554-578.

McCarthy, W. E. (2003). The REA modeling approach to teaching accounting information systems. *Issues in Accounting Education, 18*(4), 427-441.

McCarthy, W. E. (2010). *Michigan medical equipment and the other hard stuff in REA business process modeling.* Paper presented at the Semantic Modeling of Accounting Phenomena Workshop, San Francisco, CA.

McCarthy, W. E., David, J. S., & Sommer, B. S. (2003). Agility: The key to survival of the fittest in the software market. *Communications of the ACM, 46*(5), 65-69.

Murthy, U., & Wiggins, C., Jr. (2004). OOREA: An object-oriented resources, events, agents model for enterprise systems design. In *Proceedings of the International Conference on Information Systems.*

Nakamura, H., & Johnson, R. E. (1998). *Adaptive framework for the REA accounting model*. In Proceedings of the OOPSLA'98 Business Object Workshop IV.

Nittler, M. (2012). Tales of the cloud: The story of worktags. *Workday.* Retrieved from http://blogs.workday.com/story_of_worktags.html

O'Leary, D. E. (1999). REAL-D: A schema for data warehouses. *Journal of Information Systems*, *13*(1), 49-62.

O'Leary, D. E. (2004). On the relationship between REA and SAP. *International Journal of Accounting Information Systems*, *5*(1), 65-81.

O'Leary, D. E. (2015). *An architecture for integrating cloud computing and accounting and enterprise systems using REA* (working paper). University of Southern California.

Perols, J. L., & Murthy, U. S. (2012). Information fusion in continuous assurance. *Journal of Information Systems, 26*(2), 35-52.

Poels, G. (2011). Understanding business domain models: The effect of recognizing resource-event-agent conceptual modeling structures. *Journal of Database Management*, *22*(1), 69-101.

Poels, G., Maes, A., Gailly, F., & Paemeleire, R. (2011). The pragmatic quality of resources-events-agents diagrams: An experimental evaluation. *Information Systems Journal, 21*, 63-89.

Porter, M. E. (1985). *The competitive advantage: Creating and sustaining superior performance*. New York, NY: Free Press.

Rockwell, S. R. (1993). *The conceptual modeling and automated use of reconstructive accounting domain knowledge* (Doctoral Dissertation). Michigan State University.

Rockwell, S. R., & McCarthy, W. E. (1999). REACH: Automated database design integrating first-order theories, reconstructive expertise and implementation heuristics for accounting information systems. *International Journal of Intelligent Systems in Accounting, Finance and Management, 8*(3), 181-197.

Sedbrook, T. A. (2010). Maintaining enterprise knowledge with a REA-EO driven semantic wiki. *Journal of Information Technology and Information Management*, *19*(1), 19-38.

Sedbrook, T. A. (2012). Modeling the REA enterprise ontology with a domain specific language. *Journal of Emerging Technologies in Accounting, 9*, 47-70.

Sedbrook, T. A., & Newmark, R. I. (2008). Automating REA policy level specifications with semantic Web technologies. *Journal of Information Systems, 22*(2), 249-277.

Sisco, R. (2015). Chief accounting officer POV: How finance in the cloud is different. *Workday.* Retrieved from http://blogs.workday.com/chief_accounting_officer_pov_how_finance_in_the_cloud_is_different.html

Smith, J. M., & Smith, D. C. P. (1977). Database abstractions: Aggregation and generalization. *ACM Transactions on Database Systems, 2*(2), 105-133.

Sonnenberg, C., Huemer, C., Hofreiter, B., Mayrhofer, D., & Braccini, A. (2011). The REA-DSL: A domain specific modeling language for business models. In C. Mouratidis & C. Rolland (Eds.), *Advanced information systems engineering. Lecture notes in computer science* (Vol. 6741, pp. 252-266). New York: Springer.

Sorter, G. H. (1969). An "events" approach to basic accounting theory. *The* Accounting *Review, 44*(1), 12-19.

Sowa, J. (1984) *Conceptual structures: Information processing in mind and machine*. Reading, MA: Addison Wesley.

Sowa, J. (1999) *Knowledge representation: Logical, philosophical, and computational foundations*. Pacific Grove, CA: Brooks/Cole Publishing.

Sutheparaks, U., Vatanawood, W., & Patanothai, C. (2011). Defining global schema for ETL of human resource performance appraisal system using REA ontology. In *Proceedings of the 8th International Joint Conference on Computer Science and Software Engineering* (pp. 275-280).

Swete, S. (2012). Why Workday is different by design, and why it matters. *Workday.* Retrieved from http://blogs.workday.com/why_workday_is_different_by_design.html

Swete, S. (2013). Embracing continuous change at Workday. *Workday.* Retrieved from http://blogs.workday.com/embracing_continuous_change_at_workday.html

Teorey, T. J., Yang, D., & Fry, J. P. (1986). A logical design methodology for relational databases using the extended entity relationship model. *ACM Computing Surveys, 18*(2), 197-222.

Tsichritzis, D. C., & Lochovsky, F. H. (1982). *Data models*. Englewood Cliffs, NJ: Prentice-Hall.

Vandenbossche, P. E. A., & Wortmann, J. C. (2006). *Why accounting data models from research are not incorporated in ERP systems.* Paper presented at the 2nd International REA Technology Workshop, Santorini Island, Greece.

Vessey, I. (1991). Cognitive fit: A theory-based analysis of the graph versus tables literature. *Decision Sciences*, *22*(2), 219-240.

Vymetal, D., Hunka, F., Hucka, M., & Kasik, J. (2010). Enterprise-modeling based on combination of process and REA value chain perspective. *Journal of Advanced Research Management*, *2*(2), 145-158.

Wand, Y., & Weber, R. (2002). Research commentary: Information systems and conceptual modeling—a research agenda. *Information Systems Research, 13*(4), 363-376.

Weber, R. (1986). Data models research in accounting: An evaluation of wholesale distribution software. *The Accounting Review*, *61*(3), 498-518.

Weber, R. (1996). Are attributes entities? A study of database designers' memory structures. *Information Systems Research*, *7*(2), 137-162.

Weigand, H., & Elsas, P. (2012). Model-based auditing using REA. *International Journal of Accounting Information Systems*, *13*(3), 287-310.

Weigand, H., Andersson, B., Johannesson, P., Bergholtz, M., & Arachchige, J. J. (2010). Describing coordination services with REA. In M. Petit, G. Gal, A. Castiaux, J. Ralyté, & P. Plebani (Eds.), *Proceedings of the CAiSE 2010 Workshop BUSITAL'10* (pp. 120-134).

Wong, H. K. T., & Mylopoulos, J. (1977). Two views of data semantics: A survey of data models in artificial intelligence and database management. *Information 15*(3), 344-383.

Workday. (2010). Solutions: A complete suite of business management services. Retrieved from http://www.workday.com/solutions.php.

Yu, S. C. (1976). *The structure of accounting theory*. Florida: The University Presses of Florida.

Zloof, M. M. (1975). Query-by-example. *Proceedings of the National Computer Conference*, *44*, 431-438.

Zdravkovic, J., & Ilayperuma, T. (2010). Exploring REA and open-edi business frameworks for service modeling. In M. Petit, G. Gal, A. Castiaux, J. Ralyté, & P. Plebani (Eds.), *Proceedings of the CAiSE 2010 Workshop BUSITAL'10* (pp. 106-119).

Appendix

Table A1. Timeline of Influential Work

Publication	Area	Contribution	Consistency with REA and/or Influence on REA
Fisher, I. (1906). *The nature of capital and income.*	Economic theory	For exchanges that represent transfers of goods or services, value is created in a market transaction with outside parties.	Consistent with REA's definition of transfer duality and stockflow give-and-take relationships.
Black, J. D. & Black, A. G. (1929). *Production organization.*	Economic theory	Transformations create value through changes in form or substance. When resources are used, they disappear in the transformation process and lose their form so as to be unrecognizable. When resources are consumed, they are decremented in chunks that leave the original form discernible.	Consistent with REA's definition of transformation duality and stockflow production, use, and consumption relationships.
Goetz, B. E. (1939). *What's wrong with accounting?*	Accounting theory	Advocated using an unadulterated basic historical record of transactions (a theoretical precursor to databases).	Consistent with REA's database orientation.
Goetz, B. E. (1949). *Management planning and control.*	Accounting theory	Clarified and further developed Goetz (1939); advocates using a basic pecuniary record (with no accruals) plus a legal/financial supplement (in which to maintain accruals).	Consistent with REA's database orientation.
Mattesich, R. (1964). *Accounting and analytical methods.*	Accounting theory	Proposed accounting axioms that give substance to notions of economic agents and economic objects.	Influenced McCarthy's (1982) definitions of economic agents and economic resources; his duality diverged from REA's duality.
Bachman, C. W. (1965). *Integrated data store.*	Database theory	Developed beginnings of network database technology.	Gal & McCarthy (1983) built a prototype REA system in CODASYL.
Ijiri, Y. (1967). *The foundations of accounting measurement.*	Economic theory	Differentiation between causal and classificational double-entry laid the foundation for REA duality and causal networks presaged the concept of connecting REA processes into an enterprise value chain.	Influenced McCarthy's (1982) definitions of duality and connecting business processes into value chains.
McCarthy, J. & Hayes, P. (1969). *Some philosophical problems from the standpoint of artificial intelligence.*	Knowledge representation	Discussed the notions of epistemological adequacy and intensional reasoning; defines metrics for different classes of knowledge-based systems.	Influenced REA; epistemological adequacy provides the context for development of full-REA systems per Geerts & McCarthy (2000b); Intensional reasoning is pattern-matching logic; Geerts & McCarthy's (2000a) use it in their definition of claim, and it is the reason for Hessellund's (2006) concern that REA needs further specification to truly enable intensional reasoning .
Sorter, G. H. (1969). *An "events" approach to basic accounting theory.*	Accounting theory	Introduced the idea of events accounting; discussed disadvantages of value theory.	Not consistent with REA; Sorter advocates a different method of financial reporting, not a different way of accounting.

Table A1. Timeline of Influential Work

Codd, E. F. (1970). *A relational model of data for large shared data banks.*	Database theory	Introduced relational database technology.	McCarthy (1978) built a relational model for events-based accounting systems.
CODASYL (1971). *Data base task group report.*	Database theory	Introduced network database technology.	Gal & McCarthy (1983) built a prototype REA system in CODASYL.
Colantoni, C.S., Manes, R.P., & Whinston, A.B. (1971). *A unified approach to the theory of accounting and information systems.*	Accounting theory	Introduced database concepts, event coding, and key algebra.	Influenced McCarthy (1978) because they were the first to recognize the need for database management concepts in accounting, the first to propose an accounting system not based primarily on double-entry
Abrial, J. R. (1974). *Data semantics.*	Database theory	Contended that a database is a model of an evolving physical reality (i.e., all potential users of a database should identify what is important for them and the ideas should be integrated to build one conceptual data model that serves everybody).	Influenced REA, especially with respect to the semantic orientation such that the REA elements represent as directly as possible the underlying reality.
Ijiri, Y. (1975). *Theory of accounting measurement.*	Accounting theory	Emphasized accountability-driven measurement, although allowance of procedures such as periodic matching revealed a disappointing lack of full traceability.	Emphasis on accountability-driven measurement strongly influenced McCarthy's (1982) insistence on full traceability whenever possible.
Zloof, M. M. (1975). *Query-by-example.*	Database theory	Introduced the concept of Query-by-Example (QBE).	Armitage (1985) implemented a REA-oriented manufacturing system in QBE; Gal & McCarthy (1985) used QBE to implement internal controls in a REA system; Gal & McCarthy (1986) used QBE to implement a relational accounting database and query the database to derive account balances; Denna & McCarthy (1987) built a prototype relational manufacturing system integrated with decision support capabilities.
Lancaster, K. J. (1975). *Socially optimal product differentiation*.	Economic theory	Described products as bundles of attributes.	Consistent with REA; the bundles of attributes can be configured in various ways to add value to customers.
Lieberman, A. Z., & Whinston, A. B. (1975). *A structuring of an events-accounting information system.*	Accounting theory	Advocated a three part structure with user-defined database characteristics and self-organizing database capabilities.	Inconsistent with REA; their example implementations eliminated data independence and maintained the use of debits, credits, and accounts.
Chamberlin, D. D., Astrahan, M. M., Eswaran, K. P., Griffiths, P. P., Lorie, R. A., Mehl, J. W., Reisner, P., & Wade, B. W. (1976). *SEQUEL 2: A unified approach to data definition, manipulation, and control.*	Database theory	Introduced view procedures that produce dynamic windows on the database through which different classes of users may separately view information.	Influenced REA in the sense that a view that allows accountants to look at imbalances between sales and cash receipts as accounts receivable while simultaneously allowing other users to view the same objects as detailed transaction histories.

Table A1. Timeline of Influential Work

Chen, P. P. (1976). *The entity-relationship model—toward a unified view of data*	Conceptual modeling	Created the entity-relationship (ER) modeling formalism; separated the conceptual model from the physical model.	Influenced REA; McCarthy (1979) applied the ER modeling formalism to accounting and argued for semantic representation of base elements and relationships (e.g., sale-inventory and sale-customer) as opposed to syntactically representing historical accounting artifacts (e.g., journals and ledgers)
Haseman, W. D., & Whinston, A. B. (1976). *Design of a multidimensional accounting system.*	Accounting theory	Applied hierarchical database technology to organize events and defined restructuring functions.	Inconsistent with REA; their example implementations eliminated data independence and maintained the use of debits, credits, and accounts.
Yu, S. C. (1976). *The structure of accounting theory.*	Economic theory	Defined economic events as a class of phenomena that reflected changes in economic resources resulting from production, exchange, consumption, and distribution.	Influenced REA and its economic event construct.
Bubenko, J. A., Jr. (1976). *The temporal dimension in information modeling.*	Database theory	Discussed means for handling the concept of time in structured databases; discussed the notion of conclusion materialization.	Influenced REA; dealt with issues of using flow entities (events) such as sales to update stock entities (resources) such as inventory; Gal & McCarthy (1986) materialized account balances.
Everest, G. C., & Weber, R. (1977). *A relational approach to accounting models.*	Accounting theory	Introduced the ideas of data independence and normalization to accounting.	Influenced REA; suggested that future database systems should not model accounting artifacts.
Smith, J. M., & Smith, D. C. P. (1977). *Database abstractions: Aggregation and generalization.*	Conceptual modeling	Modeling of generalization hierarchies allowed much closer correspondence of system primitives with the real-world phenomena they represented.	Influenced REA; notions of aggregation and generalization are both prevalent in the REA theory.
Wong, H. K. T., & Mylopolous, J. (1977). *Two views of data semantics: A survey of data models in artificial intelligence and database management.*	Conceptual modeling	Compares and contrasts knowledge representation in the fields of database and artificial intelligence.	Influenced REA; allowed for methods of conclusion (account balance) materialization (i.e., provided procedures that one could apply to the data in the semantic model)
McCarthy, W. E. (1978). *A relational model for events-based accounting systems.*	Accounting and database theory	Applied Codd's (1970) relational database model to events accounting concepts.	Influenced REA; established database orientation.
Lum, V., Ghosh, S., Schkolnick, M., Jefferson, D., Su, S., Fry, J., Teorey, T., & Yao, B (1979). *New Orleans data base design workshop report.*	Database theory	Seminal paper on phases of database design (requirements analysis, conceptual design, logical design, physical design).	Influenced REA, especially with respect to view modeling (separately modeling each transaction cycle) and view integration (merging cycle models into one enterprise wide model).
McCarthy, W. E. (1979). *An entity-relationship view of accounting* models.	Accounting theory and conceptual modeling	Applied Chen's (1976) ER modeling formalism to accounting and semantically represented transaction data rather than syntactically representing accounting artifacts such as journals and ledgers.	Influenced REA; established semantic orientation.

Table A1. Timeline of Influential Work

McCarthy, W. E. (1980). *Construction and use of integrated accounting systems with entity-relationship modeling.*	Accounting theory and conceptual modeling	Continued to develop the ideas of representing elementary data in a way that can be shared between accountants and non-accountants.	Influenced REA; further developed database and semantic orientations.
Tsichritzis, D. C., & Lochovsky, F. H. (1982). *Data models.*	Database theory	Categorized declarative and procedural constraints; defined navigational and specificational procedures.	Influenced REA, especially with respect to the distinction between syntactic and semantic database design.
McCarthy, W. E. (1982). *The REA accounting model: A generalized framework for accounting systems in a shared data environment.*	REA	Extended his 1978, 1979, and 1980 work to include generalization hierarchies and a full structuring orientation to make it a semantic theory for an information system that tracks economic phenomena in a shared data environment without regard for ever-changing technology platforms.	Is REA.
Sowa, J. (1984). *Conceptual structures: Information processing in mind and machine.*	Knowledge representation and conceptual modeling	The definitive text on the philosophical, psychological, and linguistic foundations of conceptual modeling.	Influenced REA; concepts such as the primacy of declarative representation and conceptual relativity are evident in the Geerts & McCarthy (2000b, 2002) extensions of REA.
Porter, M. E. (1985). *The competitive advantage: Creating and sustaining superior performance.*	Value chain and strategy	The seminal text on using value chains and value systems in strategic planning.	Influenced REA; Geerts & McCarthy (1997a, 1997b, 1999, 2006) extended REA from business-process level up to the value-chain level and down to task level; Dunn (2012) also explicitly included value system level.
Hammer, M. (1990). *Reengineering work: Don't automate, obliterate.*	Strategy	Set forth principles for reengineering rather than paving the cowpaths, cementing existing processes into software, companies should start with a blank slate and determine the ideal processes to use.	Consistent with REA; rather than embedding traditional accounting artifacts into software, McCarthy started with a blank slate and determined what he believed was the best way to capture data to tell the economic story of enterprises; Andros, Cherrington, & Denna (1992) demonstrated reengineering of accounting with REA constructs in IBM's employee reimbursement system.
Gruber, T. R. (1993). *A translation approach to portable ontologies.*	Ontology	Described a mechanism for defining portable ontologies (i.e., specifications of conceptualizations: the objects, concepts, and other entities assumed to exist in an area of interest and the relationships between them).	Influenced REA; Geerts & McCarthy (2002) describe the extended REA model as a domain ontology.
Hammer, M., & Champy, J. (1993). *Reengineering the corporation.*	Value chain and strategy	Discussed the need for radical redesign of business processes to achieve improvement; linked business processes together into value chains.	Influenced REA; contributed to the expansion of REA from the business-process level to the value-chain level.

Table A1. Timeline of Influential Work

Gamma, E. Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software.*	Design patterns	Describe a framework as a set of cooperating classes that make up a reusable design for a specific class of software.	Influenced REA; structuring orientation is derived in part from the notion of reusable design patterns.
Hay, D. (1996). *Data model patterns.*	Conceptual modeling and design patterns	A comprehensive catalog of enterprise data patterns.	Consistent with the bill of materials policy-level specifications shown in Geerts & McCarthy (2006).
Fowler, M. (1997). *Analysis patterns: Reusable object models.*	Design patterns	Named the explicit representation of policies as knowledge-level representations and named the relation between the actual objects as operational-level representations	Influenced REA; Geerts & McCarthy (2002) and other studies include specification of policies via relationships between types (e.g., agent type to resource type) and include the knowledge level in addition to the operational level.
Nakamura, H., & Johnson, R. E. (1998). *Adaptive framework for the REA accounting model.*	Design patterns	Described an object-oriented framework that supports REA and overcomes limitations of relational database REA implementations.	Provides a path for future research extensions of the augmented intensional reasoning presented in Geerts & McCarthy (2000a).
Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The unified modeling language reference manual.*	Conceptual modeling	Developed a modeling language that includes several types of diagrams for various phases of system development and database design.	Influenced REA; most REA studies in the mid-2000s and later used UML class diagrams as the notation for REA models.
Sowa, J. (1999). *Knowledge representation: Logical, philosophical, and computational foundations.*	Ontology	Discussed abstract vs physical, continuants vs occurrents, firstness, secondness, and thirdness.	Influenced REA; Geerts & McCarthy (2002) use Sowa's categorization matrix to analyze REA's ontological primitives.

About the Authors

Cheryl L. Dunn is an associate professor at Grand Valley State University. She earned her PhD at Michigan State University and was a faculty member at Florida State University prior to her current position. Her research interests focus on REA accounting systems, cognitive aspects of conceptual modeling, and transformation of financial reporting. Cheryl has authored textbooks on REA accounting systems. Her research has appeared in scholarly journals such as *Decision Sciences*, *Journal of the Association for Information Systems*, *Journal of Information Systems*, *International Journal of Accounting Information Systems, Issues in Accounting Education, and Journal of Emerging Technologies in Accounting*. Cheryl has served in leadership positions in the Association for Information Systems' Sig-ASYS group and in the American Accounting Association's Accounting Information Systems and Strategic & Emerging Technologies sections. She was the book review editor for *Journal of Information Systems* and on the editorial boards of *Issues in Accounting Education*, *Journal of Database Management*, and the *International Journal of Accounting Information Systems*.

Gregory J. Gerard is an Associate Professor at Florida State University. He earned his PhD at Michigan State University. His primary research interests are conceptual modeling, and the design, use, and audit of enterprise information systems. He has published in scholarly journals such as *Journal of the Association for Information Systems, Journal of Information Systems*, *International Journal of Accounting Information Systems*, *Auditing: A Journal of Practice & Theory*, *Review of Accounting and Finance*, and *Issues in Accounting Education*. He was the president of both the Accounting Information Systems Section and the Strategic and Emerging Technologies Section of the American Accounting Association. He was the editor of a special issue of *Issues in Accounting Education* that focused on information technology. He is an associate editor *for International Journal of Accounting Information Systems* and on the editorial board of *Journal of Information Systems*.

Severin V. Grabski is an Associate Professor at Michigan State University. He earned his PhD at Arizona State University. He was also a Visiting Professor at the University of Tasmania. His current research interests include the semantic modeling of accounting phenomena, cloud computing governance, and the evaluation of the impact of ERP systems on organizations and organizational performance. He has numerous publications in scholarly journals including *MIS Quarterly*, *Decision Sciences, Journal of the Association for Information Systems, the International Journal of Accounting Information Systems, The Journal of Information Systems, Information Technology & People, Cornell Quarterly,* and *Journal of Applied Social Psychology,* He has also contributed chapters for various books including *Researching Accounting as an Information Systems Discipline*, *Enterprise Systems: ERP, Implementation and Effectiveness, Developing Quality Complex Database Systems: Practices, Techniques, Technologies,* and *Expert Systems in Developing Countries: Practice and Promise.* He is also the joint editor of the *International Journal of Accounting Information Systems*.